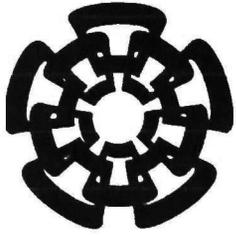


CT-814-SS1

DOV. 2015



Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional
Unidad Guadalajara

Evaluación del desempeño de localización geográfica en redes inalámbricas de sensores y actuadores

Tesis que presenta:

Joaquin Mass Sanchez

para obtener el grado de:

Maestro en Ciencias

en la especialidad de:

Ingeniería Eléctrica

Directores de Tesis

Dr. Deni Librado Torres Román

Dra. Erica Cecilia Ruiz Ibarra

**CINVESTAV
IPN
ADQUISICION
LIBROS**

CLASIF..	CT00718
ADQUIS..	CT-814-551
FECHA:	28-01-2014
PROCED.	Deu. 2014
	*

Evaluación del desempeño de localización geográfica en redes inalámbricas de sensores y actuadores

**Tesis de Maestría en Ciencias
Ingeniería Eléctrica**

Por:

Joaquin Mass Sanchez
Ingeniero en Electrónica

Instituto Tecnológico de Sonora 2006-2011

Becario de CONACYT, expediente no. 263582

Directores de Tesis

Dr. Deni Librado Torres Román

Dra. Erica Cecilia Ruiz Ibarra

Resumen

La localización es un problema en las redes de sensores inalámbricas, que consiste en determinar la posición de un nodo de interés dentro de un área geográfica definida, a partir de ciertos nodos conocidos (nodos de referencias). Este problema se ha convertido a través del tiempo en una necesidad para el hombre, ya que la información de localización es fundamental por un número de razones como: rastrear objetos móviles en un ambiente determinado, monitorear la evolución espacial de un fenómeno, determinar la calidad de cobertura de una red de sensores inalámbricos, así mismo, por un número de aplicaciones como: aprovechamiento de energía, salud, rastreo, etc.

En este trabajo se evalúa el desempeño de algoritmos de localización basados en la fuerza de la señal recibida RSS (Received Signal Strength), donde este parámetro determina la distancia entre un punto de referencia y el nodo de interés a localizar. El problema de utilizar este modelo de estimación de distancia es que es más impreciso pero es menos costoso comparado con los otros métodos existentes.

Los algoritmos de localización presentados en este trabajo son evaluados bajo dos escenarios, considerando un primer escenario en un área grande, donde los puntos de referencia son distribuidos aleatoriamente, y un segundo escenario en un área pequeña, donde los puntos de referencia son fijos. El evaluar los algoritmos de localización permitirá seleccionar aquel algoritmo más preciso y con baja complejidad computacional.

Abstract

The location is a problem in wireless sensor networks, which consists in finding the position of a node of interest, within a given geographical area, from certain known nodes (nodes of references). This problem has become over time a necessity for man, since location information is essential for a number of reasons such as: tracking moving objects in a given environment, monitor the spatial evolution of a phenomenon, identify the quality of coverage of a wireless sensor network, likewise for a number of applications such as: energy efficiency, health, tracking, etc.

In this paper we evaluate the performance of localization algorithms, based on the received signal strength (RSS), where this parameter determines the distance between a reference point and the node of interest to locate. The problem with using this model of distance estimation, is that it's vague, but it is less expensive compared to other existing methods.

Localization algorithms presented here are evaluated under two scenarios, considering a first stage in a large area where benchmarks are randomly distributed, and a second stage in a small area, where the reference points are fixed. We evaluate our localization algorithms that allow you to select more accurate algorithm with low computational complexity.

Agradecimientos

A Dios por estar cerca de mí en cada momento de mi vida.

Quiero agradecerle al CINVESTAV por haberme dado la oportunidad de haber cursado una maestría y recibir una buena educación.

Le agradezco al CONACYT por el apoyo económico.

A mis padres y hermanos por todo su apoyo y comprensión que han depositado en mí.

A mis asesores de tesis, Dr. Deni Librado Torres Román y Dra. Érica Cecilia Ruiz Ibarra, por todos sus comentarios y enseñanza durante esta etapa, así mismo agradezco al Dr. César Vargas Rosales, por haber compartido sus conocimientos, ideas durante el desarrollo de este trabajo.

A mis compañeros de generación de maestría: Laura, Gustavo, Alberto, Daniel y Benjamín, por haber compartido un año de conocimientos y experiencias.

A mis revisores de tesis, por sus comentarios y sugerencias para la redacción de este trabajo.

Índice General

Resumen	i
Abstract	ii
Agradecimientos	iii
Índice General	iv
Lista de Figuras	vi
Lista de Acrónimos	viii
1 Introducción	
1.1 Antecedentes	1
1.2 Problema	3
1.3 Objetivos	4
1.4 Justificación	4
1.5 Alcances	5
2 Técnicas de Localización en WSN	
2.1 Clasificación de las Técnicas de Localización	7
2.2 Técnicas de Estimación de Distancia	9
2.2.1 Tiempo de Llegada (ToA)	9
2.2.2 Diferencia de Tiempo de Llegada (TDoA)	9
2.2.3 Tiempo de Llegada (AoA)	10
2.2.4 Fuerza de la Señal Recibida (RSS)	11
2.2.4.1 Estimación ML de los Parámetros del Modelo Log-Normal	13
2.2.4.2 Estimador de Distancia Log-Normal	13
2.3 Técnicas de Estimación de la Posición	14
2.3.1 Trilateración	14
2.3.2 Triangulación	14
2.3.3 Multilateración por Mínimos Cuadrados Ponderados (WLS)	15
2.3.4 Algoritmo de Posicionamiento Circular	17
2.3.5 Algoritmo de Posicionamiento Hiperbólico	18
2.3.6 Algoritmo de Posicionamiento Hiperbólico Ponderado	19
2.3.7 Algoritmo de Posicionamiento Circular Ponderado	19
2.4 Técnicas de Localización Basadas en Distancia	20
2.4.1 Escalamiento Multidimensional (MDS)	20
2.4.2 Sistemas de Posicionamiento Ad-Hoc (APS)	22
2.4.2.1 Distance Vector Hop (DV-Hop)	23
2.4.2.2 Distance Vector Distance (DV-Distance)	23
2.4.2.3 Euclidian APS	24

2.5 Técnicas de Localización Libres de Distancia.....	25
2.5.1 Algoritmo de Centroide CL.....	25
2.5.1.1 Algoritmo de Centroide WCL.....	25
2.5.1.2 Algoritmo de Centroide RWL.....	26
2.5.1.3 Algoritmo de Centroide REWL.....	27
2.5.1.4 Algoritmo de Centroide con Sistemas Difusos.....	29
2.5.2 Esquema de Localización APIT.....	32
2.5.3 Intersección Rectangular.....	36
2.5.4 K Vecino más Cercano.....	36
2.5.4.1 K-Vecinos más Cercanos con Lógica Difusa.....	37
3 Evaluación de Desempeño de las Técnicas de Localización	
3.1 Parámetros del Desempeño.....	40
3.2 Descripción del Primer Escenario de Evaluación.....	41
3.3 Análisis del Desempeño de Algoritmos libres de distancia bajo escenario 1.....	42
3.4 Análisis del Desempeño de Algoritmos basados en distancia bajo escenario 1.....	48
3.5 Descripción del Segundo Escenario de Evaluación.....	51
3.6 Análisis del Desempeño de los Algoritmos Analizados bajo escenario 2.....	52
3.7 Análisis de la Complejidad Computacional.....	54
3.7.1 Complejidad Computacional medida desde un ordenador.....	55
3.7.2 Complejidad Computacional basada en el número de Operaciones Elementales.....	58
4 Algoritmo de Multilateración WLS Modificado	
4.1 Descripción del Algoritmo.....	64
4.2 Análisis de Desempeño del Algoritmo.....	67
5 Conclusiones y Trabajo Futuro	
5.1 Conclusiones.....	75
5.2 Contribuciones.....	76
5.3 Trabajo Futuro.....	77
Referencias Bibliográficas	

Lista de Figuras

1.1. Esquema de una WSN.	1
2.1. Clasificación de las técnicas de Localización.....	8
2.2. Técnica TDoA.	10
2.3. Técnica AoA.....	11
2.4. Localización del nodo móvil por la intersección de tres círculos con radios d_1, d_2 y d_3	14
2.5. Estimación de la distancia entre el nodo móvil y la estación base, utilizando AoA.	15
2.6. Esquema de Localización DV-Hop.	23
2.7. Método de Propagación Euclidiana.....	24
2.8. Escenario de evaluación para WCL.....	26
2.9. Example of Relative Span Weights.....	28
2.10. Esquema FLS.....	30
2.11. Fuzzy Membership Function of RSSI and Weight Mamdani FIS.....	31
2.12. Fuzzy Membership function of RSS Sugeno FIS.....	32
2.13. Esquema de Localización APIT.	33
2.14. Test PIT. a) Caso de adentro. b) Caso de afuera.	33
2.15. Escenarios de error para APIT test.....	34
2.16. Método SCAN para el cálculo del área de traslape.	35
2.17. Intersección Rectangular.	36
2.18. Función de pertenencia de entrada Fuzzy-KNN.....	38
2.19. Función de pertenencia de salida Fuzzy-KNN.....	38
3.1. Distribución de nodos.....	42
3.2. MSE and SD vs. (σ Noise) for node density = 1.	43
3.3. MSE and SD vs. (σ Noise) for node density = 2.	44
3.4. MSE and SD vs. (σ Noise) for node density = 5.	44
3.5. MSE vs. Node density for (σ Noise = 4 dBm).	45
3.6. SD vs. Node density for (σ Noise = 4 dBm).	45
3.7. MSE vs. Node density for (σ Noise = 1 dBm).	46
3.8. SD vs. Node density for (σ Noise = 1 dBm).	46
3.9. CDF vs. Error Location for (σ Noise = 4 dBm and Node Density = 1).	47
3.10. CDF vs. Error Location for (σ Noise = 1 dBm and Node Density = 1).	48
3.11. MSE and SD vs. (σ Noise) for Node Density = 1.	49
3.12. MSE and SD vs. (σ Noise) for Node Density = 2.	49
3.13. MSE and SD vs. (σ Noise) for Node Density = 5.	50
3.14. MSE and SD vs. Node density for (σ Noise = 1 dBm).	50

3.15. MSE and SD vs. Node density for (σ Noise = 4 dBm)	51
3.16. Escenario de Aplicación propuesto.	52
3.17. MSE vs. (σ Noise)	53
3.18. SD vs. (σ Noise)	54
3.19. CDF vs. Error Location for (σ Noise = 2)	54
3.20. Processing Time vs. Node Density for range-free algorithms.....	56
3.21. Processing Time vs. Node Density for Mamdani & Sugeno FIS, Hyperbolic and Weighted Hyperbolic algorithms.....	56
3.22. Processing Time vs. Node Density for range-based algorithms.....	58
3.23. Número de operaciones realizadas por los algoritmos basados en distancia.	61
3.24. Número de operaciones realizadas por los algoritmos libres de distancia.	62
4.1. MSE and SD vs. (σ Noise) for Node Density = 1.	67
4.2. MSE and SD vs. (σ Noise) for Node Density = 2.	68
4.3. MSE and SD vs. (σ Noise) for Node Density = 5.	68
4.4. MSE and SD vs. Node density for (σ Noise = 4 dBm).	69
4.5. MSE and SD vs. Node density for (σ Noise = 1 dBm).	70
4.6. CDF vs. Error Location for (σ Noise = 1 dBm and Node Density = 1).	70
4.7. CDF vs. Error Location for (σ Noise = 4 dBm and Node Density = 1).	71
4.8. CDF vs. Error Location for Node Density = 5.	71
4.9. CDF vs. Error Location for (σ Noise = 6 dBm and Node Density = 5).	72

Lista de Acrónimos

WSN	Wireless Sensor Network
ToA	Time of Arrival
TDoA	Time Difference of Arrival
AoA	Angle of Arrival
RSS	Received Signal Strenght
RSSI	Received Signal Strenght Indicator
WLS	Weighted Least Squares
MDS	Multidimensional Scaling
APS	System Positioning Ad-hoc
DV-Hop	Distance Vector-Hop
DV-Distance	Distance Vector-Distance
CL	Centroid Localization
WCL	Weighted Centroid Localization
RWL	Relative Span Weighted Localization
REWL	Relative Span Exponentially Weighted Localization
APIT	Approximate Point In Triangle
NOI	Node of Interest
GPS	Global Positioning System
FIS	Fuzzy Inference System
FLS	Fuzzy Logic System
MSE	Mean Squared Error
SD	Standard Desviation
KNN	K-Nearest Neighbor
CDF	Cumulative Distribution Function

Capítulo 1

Introducción

1.1 Antecedentes

Después de intensas investigaciones, las redes de sensores inalámbricos están en medio de la transición hacia el despliegue industrial en los diversos ámbitos de aplicación, tales como el automóvil, monitoreo del ambiente, la atención de la salud, la gestión de la energía, la construcción y la automatización industrial.

Una red de sensores inalámbricos WSN (Wireless Sensor Network por sus siglas en inglés) es una red con numerosos dispositivos distribuidos espacialmente, que utilizan sensores extremadamente pequeños para sensar diversas condiciones en distintos puntos, entre ellas la temperatura, el sonido, la vibración, la presión, el movimiento, etc [1]. Estos dispositivos son de poca capacidad de procesamiento, muy bajo consumo de potencia y bajo costo. Los dispositivos son unidades autónomas que constan de un microcontrolador, una fuente de energía (casi siempre una batería), un radio transceptor y un elemento sensor. En muchas aplicaciones la información sensada sólo es útil cuando es acompañada de la localización del área y las distancias aproximadas de donde se midió dicha información. Por lo tanto, los nodos sensores requieren conocer su distancia con otros sensores para estimar su posición [1, 2]. En la Figura 1 se muestra un esquema de una WSN.

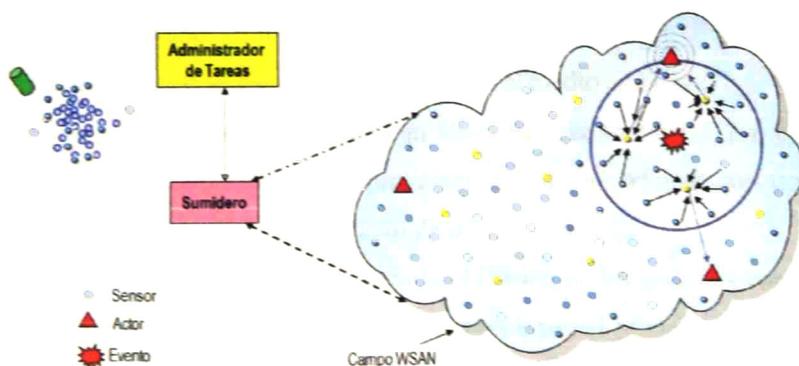


Fig. 1.1: Esquema de una WSN.

En su forma más simple, una red de sensores inalámbricos puede basarse en una estación base en la que se realiza la captura de datos que se han sentido en todos los otros dispositivos (nodos) de una manera sincronizada y estructurada. La funcionalidad de la red, es decir, las rutas y protocolos de comunicación se implementan en software y se ejecuta generalmente en microcontroladores de gama baja.

Una red de sensores inalámbrica tiene ciertas restricciones, las cuales se citan a continuación:

- Los nodos en las redes de sensores son autónomos y fuertemente desatendidos, lo cual presenta alta probabilidad de fallo.
- Los nodos de sensores son pequeños y autónomos, por lo que presentan severas restricciones energéticas, es decir la comunicación entre los nodos debe ser a través de multisalto y los nodos deben permanecer dormidos durante cierto tiempo para el ahorro del consumo de la energía.
- La cobertura de red y el ancho de banda son limitados.
- La capacidad de procesamiento y de almacenamiento de los nodos es limitada.
- Los recursos computacionales son muy limitados (memoria, CPU).

La información de la localización de los nodos en redes de sensores inalámbricas, es fundamental por un número de razones:

- Proporcionar puntos de localización.
- Localizar y rastrear la ubicación de objetos en el medio ambiente.
- Monitorear la evolución espacial de un fenómeno extraordinario. Por ejemplo, esta información es necesaria en el procesamiento de algoritmos que determinan y rastrean los límites de desplazamiento de un fenómeno.
- Determinar la calidad de cobertura. Si la ubicación de los nodos es conocida, la red puede realizar un seguimiento de la cobertura espacial proporcionada por sensores activos en cualquier momento.
- Facilitar el enrutamiento de la información a través de la red.

El problema de la localización de los nodos en WSN, puede resolverse a partir de un número de cuestionamientos clave.

1. **¿Qué significa localizar?** El término localizar en WSN se refiere a identificar la posición espacial de un nodo con coordenadas absolutas o relativas a una referencia.
2. **¿Cuánto es la precisión mínima para localizar?** Dependiendo de la aplicación se determina que técnica de localización es la más adecuada para encontrar la información de las coordenadas del nodo desconocido con la precisión requerida.
3. **¿Qué método emplear para localización?** El método de localización depende de la cantidad de parámetros de entrada que pueden ser utilizados para diferentes técnicas de localización.

La localización precisa de sensores a bajo costo es un requerimiento crítico para el despliegue de WSN en una amplia variedad de aplicaciones tales como [3, 4]: el rastreo de animales, el cual permite analizar el comportamiento animal y las interacciones con otras especies; en logística donde los sensores en equipo móvil reportan su localización cuando éste se encuentra perdido o requiere encontrarse; en la salud para determinar la ubicación de equipo y personal médico en casos de emergencias [5], entre otras aplicaciones. Entonces, la localización de los nodos desconocidos de la red, a partir de otros nodos conocidos tiene una gran aplicación, ya que pueden surgir nuevas técnicas y algoritmos más eficientes, pero sobre todo que tengan más campos de aplicación. También se pueden generar nuevos protocolos con la finalidad de obtener un algoritmo que garantice un error mínimo en la transmisión y recepción, con lo cual se podría alcanzar un gran avance en redes y telecomunicaciones.

1.2 Problema

Existen diversas aplicaciones de WSN en las cuales es útil o incluso necesario que un nodo en una red inalámbrica de sensores estime su posición geográfica. Tal es el caso del rastreo o la detección de algún evento, los cuales no son particularmente útiles si la red de sensores

no puede proporcionar información de *dónde* ha ocurrido un evento. Por lo tanto, se requieren técnicas eficientes que permitan determinar la posición de los nodos en una red durante la ocurrencia de eventos, tal que permitan desarrollar protocolos más eficientes y optimizar recursos en la red, tales como energía y tiempo de respuesta.

1.3 Objetivos

El objetivo principal de este trabajo es determinar las características de desempeño de algoritmos de localización, relacionadas con los parámetros del sistema, con la finalidad de determinar la eficiencia de éstos y la pertinencia en el uso de las redes de sensores inalámbricas. Por lo tanto los siguientes objetivos deben ser considerados:

Objetivos Específicos

- Investigar y seleccionar algoritmos de localización libres de distancia y basados en distancia.
- Investigar y proponer escenarios de evaluación del desempeño de los algoritmos de localización.
- Establecer métricas de desempeño de los algoritmos de localización, como exactitud, precisión y complejidad computacional con el fin de determinar el impacto de un algoritmo de localización ante un escenario de evaluación propuesto.
- Comparar los algoritmos de localización libres de distancia y basados en distancia mediante las métricas de desempeño ante los escenarios de evaluación propuestos, con el fin de determinar su desempeño, y el algoritmo con mejor desempeño.
- Modificar algún algoritmo de localización, con el fin de obtener un algoritmo con un mejor desempeño que los algoritmos analizados en el escenario de evaluación propuesto.

1.4 Justificación

Las técnicas de localización permiten desarrollar protocolos más eficientes y más veloces considerando las restricciones de WSN.

Al implementar técnicas de localización en diferentes escenarios en redes de sensores inalámbricos es posible conocer con precisión y exactitud la ubicación de objeto en cualquier punto ubicado dentro del marco de referencia en el cual se esté trabajando. Además, una vez realizado ésto, se tendrá un gran impacto social, ya que el conocer la ubicación de cualquier objeto implica una gran aplicación en diferentes ramas de la ingeniería como: telefonía, satélites, aeronáutica, navegación y otras ramas.

Entonces, con dichas técnicas, se podrá predecir la ocurrencia de cualquier evento; ya sea un fenómeno de índole natural, como los desastres naturales o bien algún evento inminente; también se podrá evitar la propagación de algún incidente, como incendios, o bien la ocurrencia de éste.

1.5 Alcances

Este trabajo comprende el análisis y evaluación de las técnicas de localización propuestas en la literatura, para la estimación de la posición de un objeto ubicado en un punto cualquiera; La evaluación de dichas técnicas se realiza a través de simulación en MATLAB en el cual se ejecutan diferentes pruebas de las respectivas técnicas, con el fin de identificar la técnica más eficiente. Además, en base a la evaluación mencionada, se propone un algoritmo que garantice un error mínimo y una gran exactitud y precisión en lo que respecta al punto del objeto que se desee detectar.

Capítulo 2

Técnicas de Localización en WSN

En este capítulo se presenta una clasificación de las técnicas de localización en redes de sensores inalámbricos, las cuales se dividen en dos grandes grupos: basadas en distancia (range-based) y libres de distancia (range-free). También se describen los algoritmos de localización para los dos grupos existentes.

2.1 Clasificación de las Técnicas de Localización

Actualmente existen muchas técnicas para determinar la distancia entre los nodos sensores. Los sistemas de posicionamiento global GPS (Global Positioning System) son muy útiles en exteriores, sin embargo no lo son en interiores, debido a que las señales de radio son bloqueadas por las construcciones [6, 8]. Por su parte la técnica de tiempo de llegada ToA (Time of Arrival) calcula físicamente la distancia a través de la velocidad y el tiempo de propagación de la señal, la técnica de ángulo de llegada AoA (Angle of Arrival) estima la distancia del nodo de interés NOI (Node of Interest) obteniendo la dirección de la señal enviada por nodos adjuntos a través de un arreglo de antenas y múltiples receptores, mientras que el indicador de la fuerza de la señal recibida RSSI (Received Signal Strength Indicator) mide la potencia recibida por el nodo receptor, calcula las pérdidas de propagación y las transforma a distancia, empleando un modelo teórico o empírico de las pérdidas de trayectorias de la señal [1].

Las técnicas de localización se clasifican en dos grupos: basadas en distancia (range-based) y libres de distancia (range-free) o basadas en proximidad. Las técnicas basadas en distancia, necesitan calcular la distancia entre un grupo de nodos, para estimar la posición del NOI. En este grupo, se encuentran las técnicas de multilateración [7], escalamiento multidimensional MDS (Multidimensional Scaling) [12], sistemas de posicionamiento Ad-hoc APS (System Positioning Ad-hoc) [13], algoritmos de posicionamiento circular e hiperbólico [14]. Por su parte, las técnicas libres de distancia, son aquellas, que estiman la posición del NOI mediante RSS por lo que no necesitan estimar la distancia entre los

nodos. En este grupo, están DV-Hop (Distance Vector-Hop) [7], APIT (Approximate Point In Triangle) [7, 15], centroide [7], intersección rectangular [16], intersección circular [16] e intersección hexagonal [16], entre otras.

El primer grupo necesita estimar las distancias entre el nodo desconocido y el nodo de referencia, o bien los ángulos entre ellos, para ello existen diferentes técnicas de estimación de distancia, como:

- Tiempo de llegada (ToA – Time of Arrival).
- Diferencia de tiempos de llegada (TDoA – Time Difference of Arrival).
- Angulo de llegada (AoA – Angle of Arrival).
- Fuerza de la señal recibida (RSS – Received Signal Strenght).

En la Figura 2.1 se muestra el esquema de la clasificación de las técnicas de localización. Dentro de las técnicas de localización, se mencionan las técnicas de estimación de distancia (ToA, TDoA, RSS y AoA), también se hace mención de las técnicas de localización en escenarios multi-salto, dentro de las cuales se mencionan las técnicas de localización libres de distancia (APIT, Centroide, Intersección Rectangular, etc) y basadas en distancias (MDS, Algoritmos de Posicionamiento Ad-hoc, etc), y finalmente se mencionan las técnicas de estimación de la posición (Triangulación, Trilateración y Multilateración).

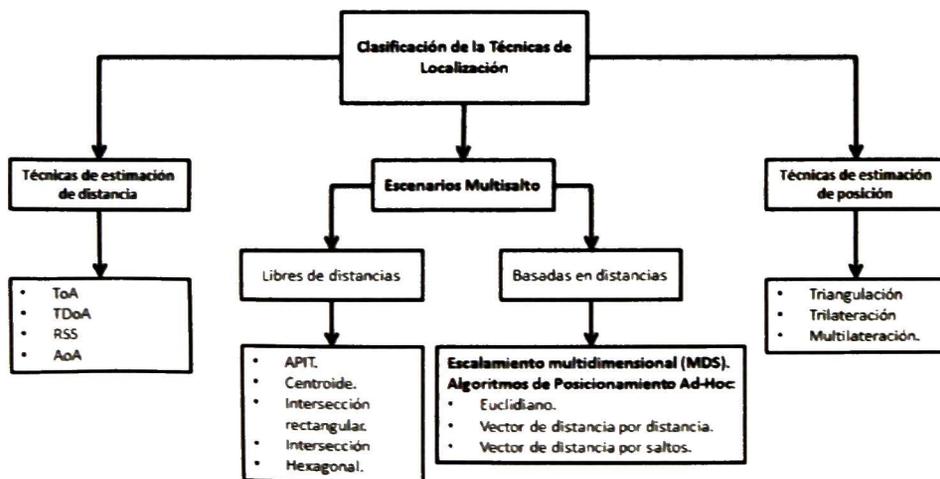


Fig. 2.1: Clasificación de las técnicas de Localización.

2.2 Técnicas de Estimación de Distancia

2.2.1 Tiempo de Llegada (ToA)

La técnica ToA explota el conocimiento de la velocidad de propagación de una señal (acústica o electromagnética) en un medio específico, para estimar la distancia entre el nodo de interés NOI y un nodo de referencia. Para estimar la posición de un nodo de interés, son requeridas al menos tres referencias. Entonces el método de trilateración basada en ToA, requiere conocimiento del tiempo de propagación de la señal; por lo tanto, la sincronización de reloj entre el NOI y un nodo de referencia es esencial para evitar errores en la estimación de la posición del NOI. Asumiendo que el NOI inicia la transmisión de una señal en un tiempo t_0 y dicha señal es recibida a un nodo de referencia $i - th$ en un tiempo t_i , entonces la distancia estimada está dada por la siguiente expresión:

$$d_i = c(t_i - t_0) \quad (2.1)$$

donde c es la velocidad de propagación de la señal ($c = 3 \times 10^8 m/s$ en el caso de una onda electromagnética en el aire).

2.2.2 Diferencia de Tiempo de Llegada (TDoA)

Esta técnica es una variante de la técnica ToA, la cual usa la diferencia en los tiempos de llegada para localizar el móvil. Esta técnica involucra la intersección de hipérbolas en lugar de círculos como se muestra en la Figura 2.2. La ventaja de este método con respecto a ToA, es que no requiere el tiempo en que el NOI inicia la transmisión, es decir t_0 , por lo tanto el tiempo de sincronización entre el NOI y las referencias no es requerido.

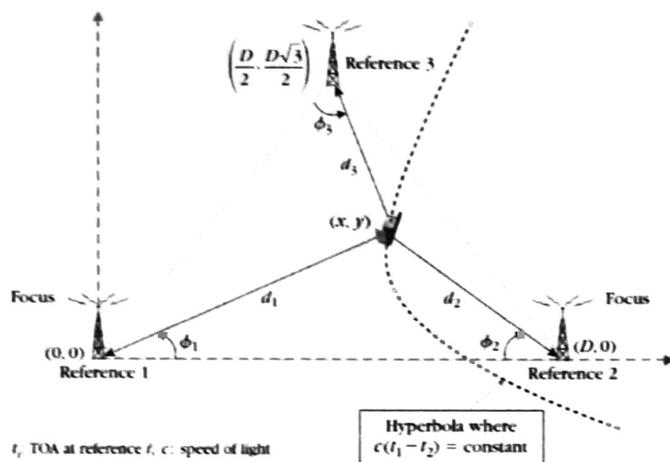


Fig. 2.2: Técnica TDoA.

La diferencia de los tiempos de llegada entre las referencias i y j ($t_i - t_j$) puede ser usado para estimar la distancia entre las referencias, la cual está dada por la siguiente expresión:

$$d_{ij} = d_i - d_j = c(t_i - t_0) - c(t_j - t_0) = c(t_i - t_j), \quad i = 1,2,3, j = 1,2,3, i \neq j \quad (2.2)$$

2.2.3 Tiempo de Llegada (AoA)

El ángulo de llegada se define como el ángulo que forma la dirección de propagación de una onda incidente y una determinada dirección de referencia, que se conoce con el nombre de orientación. La antena que posee cada sensor es utilizada para medir el ángulo de llegada.

De forma similar a ToA y TDoA, las estimaciones de AoA requieren hardware adicional demasiado caro para ser utilizado en redes de sensores muy grandes. Una ventaja de este método de estimación es que no requiere sincronización de reloj, mientras que la precisión de los ángulos es requerida para antenas direccionales.

Para estimar la posición del NOI, se necesitan al menos dos ángulos de los puntos de referencia. En la Figura 2.3 se muestra el esquema AoA.



Fig. 2.3: Técnica AoA.

2.2.4 Fuerza de la Señal Recibida (RSS)

El modelo de propagación de la señal, consiste en establecer una relación entre la atenuación de la señal y la distancia recorrida, de donde una relación entre el RSS y la distancia entre los nodos puede ser obtenida. Dependiendo de la complejidad del modelo, otros factores que pueden afectar la propagación de la señal, pueden ser considerados, por ejemplo, el tamaño de la antena o el número de obstáculos que la señal debe de atravesar.

El RSS es uno de los parámetros más populares para realizar la localización en ambientes interiores, ya que la mayoría de los dispositivos inalámbricos son capaces de medir este parámetro de forma sencilla y barata. El valor de RSS es obtenido del RSSI y es equivalente a la potencia de la señal recibida.

El RSS es el voltaje medido por el circuito RSSI por los receptores. El RSS es una medida del nivel de potencia, dado en dBm, es decir la magnitud cuadrática de la fuerza de la señal [1], en cambio RSSI es un valor adimensional arbitrario empleado para medir la fuerza de la señal otorgada por el circuito del dispositivo receptor. Sin embargo, debido a que este valor RSSI no maneja una relación directa en dBm y, a que varía de un fabricante a otro, es necesario caracterizar los valores de RSSI para asociarlo a un valor de RSS. Las mediciones de RSS son pocos costosas y simples de implementar en hardware, además de que no requieren ancho de banda o energía adicional en los sensores [3]. En este trabajo la evaluación se realiza a través de simulación, por lo que se emplea directamente RSS, en

caso de implementar dichos algoritmos será necesario medir este parámetro a través de RSSI y realizar la caracterización mencionada.

En espacio libre, la potencia de la señal es inversamente proporcional al cuadrado de la distancia de separación entre el transmisor y el receptor, por lo cual la señal se atenúa a una tasa de 20dB/década. En los sistemas de comunicaciones móviles, en modelos a gran escala, la potencia de la señal se ve afectada por reflexión, difracción y scattering. Por su parte a pequeña escala, el efecto de multitrayectorias y el fading, son los principales factores que generan error en la lectura del RSS [3, 10].

Además de las pérdidas de trayectoria, la medición de RSS es también una función de la calibración entre el transmisor y el receptor. Dependiendo del costo del proceso de manufactura, los circuitos de RSSI y la potencia del transmisor pueden variar de dispositivo a dispositivo. Además, las mediciones de potencia, pueden variar conforme se agoten las baterías, por lo que, los sensores deben ser diseñados para medir e informar sobre sus propios datos de calibración a sus vecinos. La diferencia de RSS entre dos sensores provee información acerca de la distancia relativa desde el transmisor y descarta la dependencia de la potencia de transmisión actual [3].

El modelo más típico y ampliamente usado, es el modelo log-normal, el cual es una generalización de la ecuación de espacio libre de Friis, donde la potencia decae a una tasa de $(1/d^n)$ (donde d denota la distancia o rango), y donde una variable aleatoria es añadida la cual modela los efectos de shadowing. El modelo puede ser expresado mediante la siguiente ecuación:

$$P_{RX}(dBm) = A - 10\eta \log \frac{d}{d_0} + \chi_\sigma \quad (2.3)$$

donde A es una constante, la cual denota una potencia promedio para una distancia de referencia dada por d_0 , η es el exponente de pérdidas que denota la tasa de decaimiento de la potencia con respecto a la distancia y χ es una variable aleatoria Gaussiana con media cero y varianza σ_χ^2 que está definida en dBs. La ecuación del modelo log-normal es una

línea con pendiente 10η cuando es trazada con respecto a la distancia. El rango típico del exponente de pérdidas está entre valores de 1.5 y 5. Típicamente, el valor de σ es tan bajo como 4 y tan alto como 12 [6, 10].

2.2.4.1 Estimación ML de los Parámetros del Modelo Log-Normal

Los parámetros η y σ_χ^2 son usualmente desconocidos y deben ser estimados basado sobre las mediciones del canal. Considerando un conjunto de N observaciones de potencias promedios obtenidas a diferentes distancias de un transmisor y diferentes puntos de localización en el área de interés tal que N pares de mediciones $\{P_{RX}(d_i), d_i\}$ son disponibles. Por simplicidad se asumen que las potencias están dadas en dB. La función log-likelihood es obtenida de la función de densidad de probabilidad conjunta de las N observaciones:

$$L(n, \sigma_\chi^2) = -N \ln \left(\sqrt{2\pi\sigma_\chi^2} \right) - \frac{1}{2\sigma_\chi^2} \sum_{i=1}^N \left[P_{RX}(d_i) - P_{RX}(d_0) - 10\eta \log \left(\frac{d_i}{d_0} \right) \right]^2 \quad (2.4)$$

Minimizando la expresión anterior con respecto a η , entonces la estimación está dada por:

$$\hat{\eta} = \frac{\sum_{i=1}^N \log \left(\frac{d_i}{d_0} \right) [P_{RX}(d_i) - P_{RX}(d_0)]}{10 \sum_{i=1}^N \left(\log \left(\frac{d_i}{d_0} \right) \right)^2} \quad (2.5)$$

Sustituyendo η por $\hat{\eta}$ en la ecuación (2.4) y derivando con respecto a σ_χ^2 , se obtiene la siguiente expresión:

$$\sigma_\chi^2 = \frac{1}{N} \sum_{i=1}^N \left[P_{RX}(d_i) - P_{RX}(d_0) - 10\hat{\eta} \log \left(\frac{d_i}{d_0} \right) \right]^2 \quad (2.6)$$

2.2.4.2 Estimador de Distancia Log-Normal

Cuando los parámetros del canal han sido estimados, las mediciones de la potencia pueden ser utilizadas para estimar la distancia entre el transmisor y el receptor. Sustituyendo los valores de $\hat{\eta}$ y $\hat{\sigma}_\chi^2$, y redefiniendo la ecuación (2.4), para una simple observación de potencia P_{RX} , se obtiene la siguiente expresión:

$$L(\hat{n}, \hat{\sigma}_\chi^2, d) = -N \ln \left(\sqrt{2\pi\hat{\sigma}_\chi^2} \right) - \frac{1}{2\hat{\sigma}_\chi^2} \left[P_{RX}(d) - P_{RX}(d_0) - 10\hat{\eta} \log \left(\frac{d}{d_0} \right) \right]^2 \quad (2.7)$$

Derivando la expresión (2.4) con respecto a d , se obtiene la expresión del estimador de distancia:

$$\hat{d} = d_0 10^{\frac{P_{RX}(d_0) - P_{RX}}{10\eta}} \quad (2.8)$$

2.3 Técnicas de Estimación de la Posición

2.3.1 Trilateración

Este método utiliza la distancia entre una estación base y el nodo móvil. Requiere al menos 3 estaciones base. La localización del nodo móvil es determinada usando una técnica de localización de detección de RF.

El dispositivo móvil es localizado por la intersección de tres círculos. Cada radio del círculo, representa la distancia entre el dispositivo móvil y la estación base considerada como se ilustra en la Figura 2.4

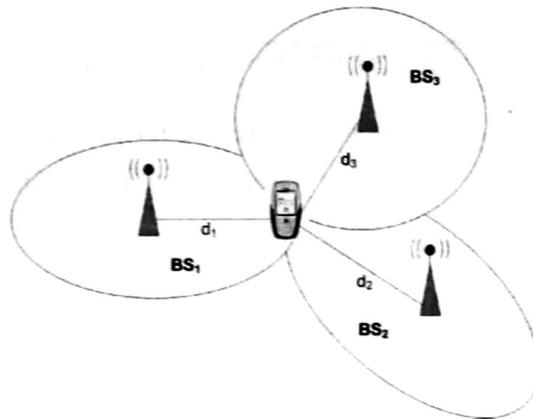


Fig. 2.4: Localización del nodo móvil por la intersección de tres círculos con radios d_1 , d_2 y d_3 .

2.3.2 Triangulación

Este método está basado en la estimación del AoA de una señal del dispositivo móvil a ser localizado. En la Figura 2.5 se observa un ejemplo del método de posicionamiento por triangulación.

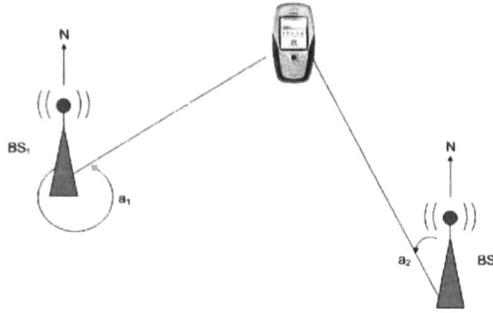


Fig. 2.5: Estimación de la distancia entre el nodo móvil y la estación base, utilizando AoA.

2.3.3 Multilateración por Mínimos Cuadrados Ponderados (WLS)

En presencia del ruido no deseado, un rango entre $i - th$ nodos de referencia al NOI, puede ser expresado mediante la siguiente expresión [7]:

$$r_i = f_i(x, y) + n_i \tag{2.9}$$

donde:

$$f_i(x, y) = \sqrt{(x - x_i)^2 + (y - y_i)^2} \tag{2.10}$$

Recolectando N mediciones de varios nodos de referencia, en un vector de observación $r = [r_1 \dots r_N]$, se obtiene la siguiente expresión:

$$r = f(x, y) + n \tag{2.11}$$

donde $f(x, y) = [f_1(x, y) \dots f_N(x, y)]^H$ es un vector que colecta las mediciones libres de ruido entre el nodo de referencia y el NOI, y $\eta = [n_1 \dots n_N]^H$ es un vector de ruido, con media cero, y matriz de covarianza $K = E\{\eta\eta^H\}$.

Para resolver este problema, se empieza definiendo un vector bidimensional q , el cual contiene las coordenadas del nodo de interés, es decir $q = [x \ y]^H$, entonces, dado un set de observaciones de ruido $r = f(x, y) + n$, el problema de optimización por mínimos cuadrados (LS) se propone de la siguiente manera:

$$\hat{q} = \underset{q}{\operatorname{argmin}} [r - f(q)]^H [r - f(q)] \tag{2.12}$$

donde la matriz de covarianza es conocida, entonces, una variante del problema LS es llamado WLS (Weighted Least Squares), el cual es propuesto mediante la siguiente expresión:

$$\hat{\mathbf{q}} = \underset{\mathbf{q}}{\operatorname{argmin}} [\mathbf{r} - \mathbf{f}(\mathbf{q})]^H \mathbf{K}^{-1} [\mathbf{r} - \mathbf{f}(\mathbf{q})] \quad (2.13)$$

donde la función de costo está definida mediante la siguiente expresión:

$$J = [\mathbf{r} - \mathbf{f}(\mathbf{q})]^H \mathbf{K}^{-1} [\mathbf{r} - \mathbf{f}(\mathbf{q})] \quad (2.14)$$

De las ecuaciones (2.13) y (2.14) se observa que describen un problema no lineal. Entonces este problema puede ser resuelto por numerosas técnicas de minimización, como interior-reflective Newton algorithm. La solución de este tipo de problemas puede presentar varios inconvenientes, como costo computacional, o bien convergencia causados por puntos con mínimo local. Para evitar estos problemas surge un proceso de linealización, basado en la expansión de las series de Taylor de primer orden del vector función $\mathbf{f}(\mathbf{q})$ considerando un punto conocido $\mathbf{q}_0 = [x_0 y_0]^H$

Considerando \mathbf{q}_0 suficientemente cercano a la posición real del NOI, es decir \mathbf{q} , las series de Taylor están dadas mediante la siguiente expresión:

$$\mathbf{f}_l(\mathbf{q}) = \mathbf{f}(\mathbf{q}_0) + \mathbf{D}(\mathbf{q} - \mathbf{q}_0) \quad (2.15)$$

donde $\mathbf{f}_l(\mathbf{q})$ es la función linealizada y \mathbf{D} es una matriz de dimensión $N \times 2$, dada por:

$$\mathbf{D} = \begin{bmatrix} \left. \frac{\partial f_1(\mathbf{q})}{\partial x} \right|_{\mathbf{q}=\mathbf{q}_0} & \left. \frac{\partial f_1(\mathbf{q})}{\partial y} \right|_{\mathbf{q}=\mathbf{q}_0} \\ \vdots & \vdots \\ \left. \frac{\partial f_N(\mathbf{q})}{\partial x} \right|_{\mathbf{q}=\mathbf{q}_0} & \left. \frac{\partial f_N(\mathbf{q})}{\partial y} \right|_{\mathbf{q}=\mathbf{q}_0} \end{bmatrix} \quad (2.16)$$

donde:

$$\left. \frac{\partial f_i(\mathbf{q})}{\partial x} \right|_{\mathbf{q}=\mathbf{q}_0} = \frac{x_0 - x_i}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}}, \quad i = 1, 2, \dots, N \quad (2.17)$$

$$\left. \frac{\partial f_i(\mathbf{q})}{\partial y} \right|_{\mathbf{q}=\mathbf{q}_0} = \frac{y_0 - y_i}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}}, \quad i = 1, 2, \dots, N \quad (2.18)$$

Sustituyendo (2.15) en (2.14) y redefiniendo el vector de observación como:

$$\mathbf{r}' = \mathbf{r} - \mathbf{f}(\mathbf{q}_0) + \mathbf{D}\mathbf{q}_0 \quad (2.19)$$

Entonces la nueva función de costo queda expresada por:

$$\mathbf{J} = [\mathbf{r}' - \mathbf{D}\mathbf{q}]^H \mathbf{K}^{-1} [\mathbf{r}' - \mathbf{D}\mathbf{q}] \quad (2.20)$$

La cual se puede observar que es lineal, comparado con la ecuación (2.14). Entonces, minimizando la función de costo (2.20) con respecto a \mathbf{q} , el estimador WLS queda expresado por:

$$\hat{\mathbf{q}} = (\mathbf{D}^H \mathbf{K}^{-1} \mathbf{D})^{-1} \mathbf{D}^H \mathbf{K}^{-1} \mathbf{r}' \quad (2.21)$$

Sustituyendo (2.19) en (2.21), el estimador WLS puede ser reescrito como

$$\hat{\mathbf{q}} = \mathbf{q}_0 + (\mathbf{D}^H \mathbf{K}^{-1} \mathbf{D})^{-1} \mathbf{D}^H \mathbf{K}^{-1} [\mathbf{r} - \mathbf{f}(\mathbf{q}_0)] \quad (2.22)$$

Observando la ecuación (2.22), el estimador WLS puede ser encontrado con una recursión, donde en cada iteración la posición actual $\hat{\mathbf{q}}$ se utiliza como \mathbf{q}_0 para la siguiente iteración. Entonces, la recursión está dada por [28]:

$$\hat{\mathbf{q}}(\mathbf{n} + 1) = \hat{\mathbf{q}}(\mathbf{n}) + (\mathbf{D}^H \mathbf{K}^{-1} \mathbf{D})^{-1} \mathbf{D}^H \mathbf{K}^{-1} [\mathbf{r} - \mathbf{f}(\hat{\mathbf{q}}(\mathbf{n}))] \quad (2.23)$$

2.3.4 Algoritmo de Posicionamiento Circular

La idea de este algoritmo es encontrar la posición del nodo móvil, que minimiza la suma de los errores cuadráticos en el conjunto de distancias estimadas [14]. Si (x_i, y_i) es la posición del nodo de referencia i , y \hat{d}_i es la distancia estimada a un nodo de referencia, entonces el error cuadrático medio MSE (Mean Squared Error) está dado por la siguiente expresión:

$$\varepsilon = \sum_{i=1}^N (\sqrt{(x_i - x)^2 + (y_i - y)^2} - \hat{d}_i)^2 \quad (2.24)$$

donde N es el número de nodos de referencia. La posición del nodo móvil puede ser calculada iterativamente con el método del gradiente descendiente:

$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}_{k+1} = \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}_k - \alpha \begin{bmatrix} \frac{\partial \varepsilon}{\partial x} \\ \frac{\partial \varepsilon}{\partial y} \end{bmatrix}_{x=\hat{x}_k, y=\hat{y}_k} \quad (2.25)$$

Este método requiere de una posición inicial, la cual puede ser el centroide de las posiciones de los nodos de referencia.

2.3.5 Algoritmo de Posicionamiento Hiperbólico

El algoritmo de posicionamiento hiperbólico convierte el problema no lineal en un problema lineal, con un estimador de mínimos cuadrados [14]. La distancia entre un nodo móvil y un nodo de referencia i , puede ser expresada por.

$$d_i^2 = (x_i - x)^2 + (y_i - y)^2. \quad (2.26)$$

Desarrollando la expresión anterior, se llega a lo siguiente.

$$2xx_i + 2yy_i = x_i^2 + y_i^2 - d_i^2 + d_1^2. \quad (2.27)$$

Expresando (2.27) en forma matricial.

$$\begin{bmatrix} 2x_2 & 2y_2 \\ \vdots & \vdots \\ 2x_n & 2y_n \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x_2^2 + y_2^2 - d_2^2 + d_1^2 \\ \vdots \\ x_n^2 + y_n^2 - d_n^2 + d_1^2 \end{bmatrix}. \quad (2.28)$$

Entonces el problema puede ser formulado por.

$$\mathbf{H}\hat{\mathbf{p}} = \tilde{\mathbf{b}}. \quad (2.29)$$

Donde $\mathbf{H} = \begin{bmatrix} 2x_2 & 2y_2 \\ \vdots & \vdots \\ 2x_n & 2y_n \end{bmatrix}$, $\hat{\mathbf{p}} = \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}$ y $\tilde{\mathbf{b}}$ es un vector aleatorio dado por.

$$\tilde{\mathbf{b}} = \begin{bmatrix} x_2^2 + y_2^2 - \tilde{d}_2^2 + \tilde{d}_1^2 \\ \vdots \\ x_n^2 + y_n^2 - \tilde{d}_n^2 + \tilde{d}_1^2 \end{bmatrix}. \quad (2.30)$$

Finalmente, la posición del nodo móvil puede ser calculada por la siguiente expresión.

$$\hat{\mathbf{p}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \tilde{\mathbf{b}}. \quad (2.31)$$

donde $\hat{\mathbf{p}}$ es la posición estimada del NOI.

2.3.6 Algoritmo de Posicionamiento Hiperbólico Ponderado

El método tradicional de posicionamiento hiperbólico calcula la posición del nodo móvil, mediante la expresión (2.31). Por lo tanto, este problema lineal puede ser resuelto usando un estimador de mínimos cuadrados ponderados propuesto en [29]. En este caso, la posición del nodo móvil puede ser estimada por:

$$\mathbf{s} = (\mathbf{A}^T \mathbf{S}^{-1} \mathbf{A})^{-1} \mathbf{A}^T \mathbf{S}^{-1} \mathbf{b} \quad (2.32)$$

Donde, \mathbf{S} es la matriz de covarianza del vector \mathbf{b} , la cual se expresa por:

$$\mathbf{S} = \begin{bmatrix} 1 + \left(\frac{d_2}{d_1}\right)^4 & 1 & \dots & 1 \\ 1 & 1 + \left(\frac{d_2}{d_1}\right)^4 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 + \left(\frac{d_N}{d_1}\right)^4 \end{bmatrix} \quad (2.33)$$

Como se observa en la expresión anterior, los elementos de la matriz de covarianza \mathbf{S} dependen de la distancia real entre el móvil y el NOI. Por lo tanto, con el fin de utilizar el estimador en (2.32) en una implementación real, es necesario aproximar las distancias reales d_i por las distancias estimadas \hat{d}_i .

2.3.7 Algoritmo de Posicionamiento Circular Ponderado

Este algoritmo es similar al algoritmo de posicionamiento circular, sólo que se le agrega un factor de peso el cual contribuye a que las distancias medidas sean más precisas, es decir con una varianza más pequeña. El problema se puede simplificar en la siguiente expresión, la cual consiste en encontrar las coordenadas (x, y) que minimizan la expresión:

$$\varepsilon = \frac{1}{d_i^2} \sum_{i=1}^N (\sqrt{(x_i - x)^2 + (y_i - y)^2} - \hat{d}_i)^2 \quad (2.34)$$

Como se observa en la ecuación (2.34), el factor de peso depende de la distancia real d_i entre el nodo móvil y el nodo de referencia i . Entonces, al utilizar este método en un ambiente real, es necesario aproximar la distancia real d_i a la distancia estimada \hat{d}_i . La posición del nodo móvil puede ser calculada utilizando el método del gradiente descendiente (2.25).

2.4 Técnicas de Localización Basadas en Distancia

2.4.1 Escalamiento Multidimensional (MDS)

Este método es usado para el análisis de disimilitud de datos sobre un conjunto de objetos, puede descubrir la estructura espacial en el dato [12]. La ventaja de este método es que siempre genera alta precisión en la estimación de la posición.

Se asume que se tienen n objetos con disimilaridades δ_{rs} . Esta técnica MDS, intenta encontrar un conjunto de puntos en un espacio donde cada punto representa uno de los objetos y las distancias entre los puntos d_{rs} son tales que $d_{rs} \approx f(\delta_{rs})$, donde f es una función que intenta transformar las disimilaridades en una forma de distancia.

Sea $T = [t_{ij}]_{n \times 2}$, la cual denota la verdadera localización del conjunto de n nodos sensores en un espacio 2-D. $d_{ij}(T)$ es la distancia entre los sensores i y j basado sobre su posición en T :

$$d_{ij}(T) = (\sum_{a=1}^m (t_{ia} - t_{ja})^2)^{1/2} \quad (2.35)$$

Donde:

$T = [t_{ij}]_{n \times 2}$, denota las posiciones reales del conjunto de n sensores.

δ_{ij} = distancia entre los sensores i e j . Si no hay errores, entonces $\delta_{ij} = d_{ij}(T)$.

En resumen, si todos los pares de distancias de los sensores en T son recogidos, se puede utilizar el algoritmo de escalamiento multidimensional clásico (MDS) para estimar la posición de los sensores. El algoritmo consta de los siguientes pasos:

- 1) Calcular la matriz de distancias cuadradas D^2 , donde $D = [d_{ij}]_{n \times n}$.
- 2) Calcular la matriz J con $J = I - e * e^T / n$, donde $e = [1, 1, \dots, 1]$.
- 3) Aplicar doble centrado a la matriz J con $H = -\frac{1}{2} J D^2 J$.
- 4) Calcular los eigenvalores mediante la descomposición espectral $H = UVU^T$
- 5) Suponiendo que se desean obtener i dimensiones de la solución, ($i = 2$) en el caso de 2D, entonces se denota la matriz de los eigenvalores más grandes como V_i y U_i como las primeras i columnas de U . Entonces, la coordenada del NOI mediante este método, está dada por:

$$X = U_i V_i^{1/2} \tag{2.36}$$

Con este método se obtienen las coordenadas relativas del NOI, por lo que es necesario alinear esta coordenada con los nodos de referencia, para obtener las coordenadas físicas del NOI. A continuación se presenta un ejemplo numérico del MDS clásico para describir el método.

Considerando la siguiente matriz de distancia:

$$D = \begin{bmatrix} 0 & 569 & 667 & 530 \\ 569 & 0 & 1212 & 1043 \\ 667 & 1212 & 0 & 201 \\ 530 & 1043 & 201 & 0 \end{bmatrix}$$

Calculando la matriz distancias cuadradas D^2 y calculando la matriz J , donde $n = 4$. Puesto que D es una matriz 4×4 , entonces:

$$\begin{aligned} J = I - e * \frac{e^T}{n} &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \frac{1}{4} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \times [1 \quad 1 \quad 1 \quad 1] \\ &= \begin{bmatrix} 3/4 & -1/4 & -1/4 & -1/4 \\ -1/4 & 3/4 & -1/4 & -1/4 \\ -1/4 & -1/4 & 3/4 & -1/4 \\ -1/4 & -1/4 & -1/4 & 3/4 \end{bmatrix} \end{aligned}$$

Aplicando doble centrado a la matriz J se obtiene la matriz H .

$$H = -\frac{1}{2}JD^2J = \begin{bmatrix} 34466 & 101461 & -74893 & -61034 \\ 101461 & 492217 & -358045 & -235633 \\ -74893 & -358045 & 260637 & 172301 \\ -61034 & -235633 & 172301 & 124366 \end{bmatrix}$$

Finalmente se realiza la descomposición SVD de la matriz H .

$$U = \begin{bmatrix} -0.1611 & 0.7287 & 0.4393 & 0.5000 \\ -0.7419 & -0.3180 & -0.3136 & 0.5000 \\ 0.5404 & 0.1711 & -0.6548 & 0.5000 \\ 0.3627 & -0.5818 & 0.5291 & 0.5000 \end{bmatrix},$$

$$V = \begin{bmatrix} 8.9021e5 & 0 & 0 & 0 \\ 0 & 0.2133e5 & 0 & 0 \\ 0 & 0 & 0.0014e5 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

En este caso, como las coordenadas del NOI son en 2D, se seleccionan los 2 eigenvalores más grandes de la matriz V , los cuales están en la columna 1 y 2, entonces se escogen las columnas 1 y 2 de la matriz U .

$$X = U_2V_2^{1/2} = \begin{bmatrix} -0.1611 & 0.7287 \\ -0.7419 & -0.3180 \\ 0.5404 & 0.1711 \\ 0.3627 & -0.5818 \end{bmatrix} \begin{bmatrix} 9.43e2 & 0 \\ 0 & 1.46e2 \end{bmatrix} = \begin{bmatrix} -152.028 & -106.422 \\ -700.028 & 46.445 \\ 509.854 & -24.995 \\ 342.202 & 84.972 \end{bmatrix}$$

Finalmente, se puede notar que la suma de las columnas de la matriz X es igual a cero, por lo que las coordenadas relativas es el centroide de la configuración de todos los puntos.

2.4.2 Sistemas de Posicionamiento Ad-Hoc (APS)

No es deseable tener nodos de referencias que emitan señales con una amplia cobertura de radio, debido a colisiones, gran consumo de potencia y problema de cobertura cuando éstos están en movimiento. Para esto se utiliza el modelo de propagación por saltos. La diferencia de este método con el GPS es que los nodos de referencia son conectados por saltos en lugar de una conexión directa. APS utiliza una versión simplificada de GPS, trabajando únicamente con distancias, sin necesidad de relojes para la sincronización. El requisito fundamental para que este sistema sea operativo es que en la red haya al menos tres balizas.

2.4.2.3 Euclidian APS

Este tercer método es el más parecido al GPS, el cual involucra la distancia euclidiana real a un nodo de referencia. En la Figura 2.7 se muestra dicho método.

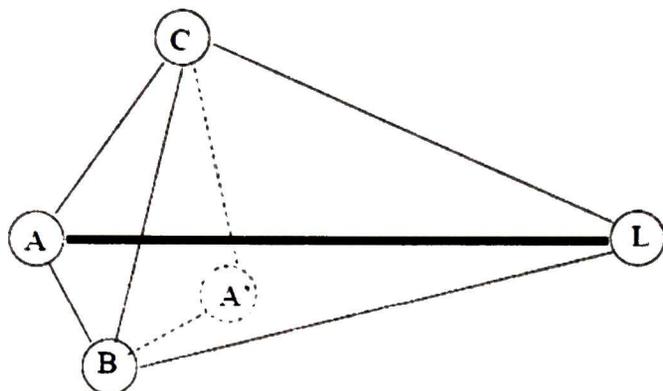


Fig. 2.7: Método de Propagación Euclidiana.

Como se muestra en la Figura 2.7, L es el nodo de referencia, B y C son los nodos vecinos, quienes fueron estimados por L, A es un nodo arbitrario que necesita tener al menos dos vecinos B y C.

En cualquier caso para el cuadrilátero ABCL todos los lados son conocidos, y una de las diagonales, es decir BC. Esto permite que el nodo A calcule la segunda diagonal AL, cual representa la distancia euclidiana entre el nodo A y la referencia L. Es posible que el nodo A esté del mismo lado de BC como lo está la referencia L, el cual se muestra como A' en la figura 2.7; en este caso la distancia a la referencia L es diferente. El escoger entre dos posibilidades es hecha localmente por A, cualquiera por votación, cuando A tiene varios nodos vecinos cercanos estimados por L, o examinando la relación con otros vecinos comunes de B y C. Si no puede escoger con claridad entre A y A', una de las distancias a L no será la correcta para A hasta que cualquier otro vecino sea estimado por la referencia L o que más vecinos en un segundo salto sean estimados por L, así se podrá escoger con claridad. Una vez que se haya escogido A, la distancia actual estimada es calculada mediante el teorema de Pitágoras, generalizado en los triángulos ACB, BCL y ACL, para encontrar la distancia AL.

2.4.2.3 Euclidian APS

Este tercer método es el más parecido al GPS, el cual involucra la distancia euclidiana real a un nodo de referencia. En la Figura 2.7 se muestra dicho método.

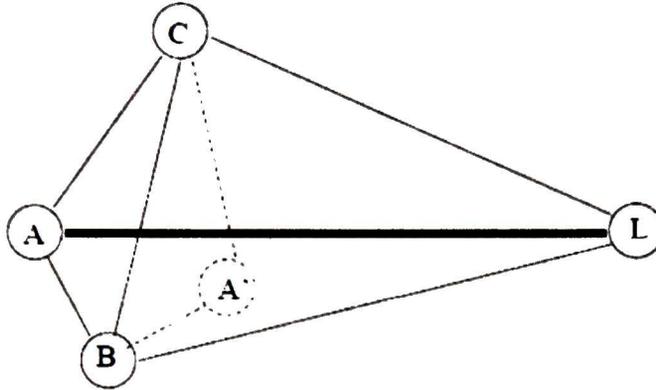


Fig. 2.7: Método de Propagación Euclidiana.

Como se muestra en la Figura 2.7, L es el nodo de referencia, B y C son los nodos vecinos, quienes fueron estimados por L, A es un nodo arbitrario que necesita tener al menos dos vecinos B y C.

En cualquier caso para el cuadrilátero ABCL todos los lados son conocidos, y una de las diagonales, es decir BC. Esto permite que el nodo A calcule la segunda diagonal AL, cual representa la distancia euclidiana entre el nodo A y la referencia L. Es posible que el nodo A esté del mismo lado de BC como lo está la referencia L, el cual se muestra como A' en la figura 2.7; en este caso la distancia a la referencia L es diferente. El escoger entre dos posibilidades es hecha localmente por A, cualquiera por votación, cuando A tiene varios nodos vecinos cercanos estimados por L, o examinando la relación con otros vecinos comunes de B y C. Si no puede escoger con claridad entre A y A', una de las distancias a L no será la correcta para A hasta que cualquier otro vecino sea estimado por la referencia L o que más vecinos en un segundo salto sean estimados por L, así se podrá escoger con claridad. Una vez que se haya escogido A, la distancia actual estimada es calculada mediante el teorema de Pitágoras, generalizado en los triángulos ACB, BCL y ACL, para encontrar la distancia AL.

2.5 Técnicas de Localización Libres de Distancia

Las técnicas de localización libres de distancia involucran la fuerza de la señal recibida para estimar la posición del NOI. Dentro de este grupo, existen algunas técnicas que sólo necesitan la posición de los nodos de referencia para estimar la posición de un nodo, por ejemplo el método APIT, el cual crea todas las combinaciones posibles de triángulos con los nodos de referencia para estimar el área de traslape de dichos triángulos en donde se encuentra el NOI. La ventaja de este método es que es muy preciso para redes de alta densidad de nodos pero su costo computacional es muy alto. También existen otros métodos basados en el mismo criterio que APIT, por ejemplo el algoritmo de intersección circular, el cual consiste en encontrar el área de intersección de círculos formados por los nodos de referencia. Este método también es computacionalmente muy costoso. Sin embargo el algoritmo de intersección rectangular requiere menos complejidad computacional, ya que la figura resultante de la intersección de cuadros es un rectángulo, en cambio la figura formada por la intersección de círculos es más compleja; pero este algoritmo es más impreciso que el algoritmo de intersección circular [16].

2.5.1 Algoritmo de Centroide CL

El método CL (Centroid Localization) no utiliza la potencia de la señal, para determinar la distancia o el AoA de los nodos de referencias al NOI, entonces, para calcular la posición del NOI, este nodo recibe las coordenadas de los nodos de referencias y el NOI simplemente estima el centroide de dichos puntos [17], el cual está dado por:

$$\hat{p} = \frac{1}{n} \sum_{i=0}^n p_i \quad (2.38)$$

Donde \hat{p} es la coordenada del NOI y $p_i = (x_i, y_i)$ es la coordenada de la referencia i .

2.5.1.1 Algoritmo de Centroide WCL

El método de centroide CL asume que todos los puntos están a la misma distancia del NOI. Sin embargo, [18] toma en cuenta que algunos puntos pueden estar más cerca que otros al NOI, bajo este argumento propone WCL (Weighted Centroid Localization), en el cual se toma en cuenta la potencia de la señal para estimar la distancia entre el NOI y los nodos de referencia. En este método algunos puntos están más cercanos al nodo de interés, por lo

tanto el NOI recibe las distancias de los nodos de referencias y los pesos, los cuales dependen de la distancia y del parámetro g . En (2.39) se muestra la ecuación que estima la posición del NOI y la expresión del factor de ponderación de la referencia i sobre el NOI.

$$\hat{p} = \frac{\sum_{i=0}^n w_i p_i}{\sum_{i=0}^n w_i} \quad \text{con} \quad w_i = \frac{1}{(d_i)^g} \quad (2.39)$$

donde d_i es la distancia conocida, estimada por algún método como ToA, TDoA, AoA o RSS entre el nodo desconocido al nodo de referencia p_i . Observando la ecuación (2.39), el factor de peso w_i es mayor para distancia pequeñas y pequeño para distancias grandes, es decir, si el NOI recibe muy baja potencia de algún nodo de referencia, entonces el peso es muy pequeño, en algunos casos puede ser despreciable. Los valores de g son determinados manualmente acorde a [19], por ejemplo para un rango de transmisión de 10m, $g = 1$ y para un rango de transmisión de 20m, $g = 2$.

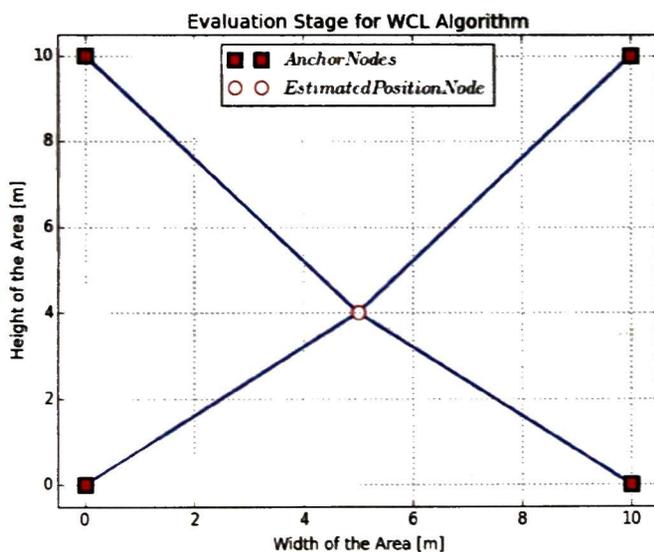


Fig. 2.8. Escenario de evaluación para WCL.

2.5.1.2 Algoritmo de Centroide RWL

Este método RWL (Relative Span Weighted Localization) es similar al método WCL, sólo que los pesos son asignados mediante la relación lineal de los valores RSS medidos de cada receptor R_i sobre el rango RSS [20], definido como:

$$v^{\Delta} = v_{max} - v_{min} \quad (2.40)$$

donde:

$$v_{max} = \max\{v_i \in \gamma\}$$

$$v_{min} = \min\{v_i \in \gamma\}$$

$$\gamma = \{v_i; v_i \text{ es el valor de RSS para el mensaje } \mathcal{M}_T \text{ a } R_i \text{ para todo } R_i \in \mathbb{R}\}$$

Definiendo el valor mínimo y máximo de RSS, v_{max} es el máximo valor de RSS y v_{min} es el mínimo valor de RSS para el mensaje \mathcal{M}_T . El rango del conjunto de valores de RSS medidos para cada receptor $R_i \in \mathbb{R}$ se define como v^{Δ}

El peso w_i de cada receptor R_i , es calculado mediante la siguiente expresión:

$$w_i = \frac{v_i - v_{min}}{v^{\Delta}} \quad (2.41)$$

El RWL centroide es calculado mediante la siguiente expresión:

$$\hat{p} = \frac{\sum_{i=1}^n (v_i - v_{min}) p_i}{\sum_{i=1}^n (v_i - v_{min})} \quad (2.42)$$

donde \hat{p} es la coordenada estimada del NOI.

2.5.1.3 Algoritmo de Centroide REWL

El algoritmo REWL (Relative Span Exponentially Weighted Localization) es una variante al WCL en donde los pesos aumentan exponencialmente con la fuerza del RSS [20], de esta manera se acentúa más el efecto de los nodos más cercanos en el cálculo del centroide. En el método REWL se propone un factor de peso λ , el cual es un factor de ajuste de un peso w_i . Este parámetro de control λ , toma valores reales positivos entre 0 y 1. Por ejemplo, si $\lambda > 1$, entonces el factor $(1 - \lambda)$ es negativo, lo cual influye en que el peso w_i , se comporte como una curva con oscilaciones, es decir el factor w_i estaría variando de positivo a negativo, lo cual no está bien, ya que a mayor potencia mayor peso y menor potencia menor peso. Si $(\lambda = 0,1)$, para estos dos valores, el factor $(1 - \lambda)$ toma valores de 1,0, por lo tanto el peso w_i toma valores constantes, lo cual no es válido, ya que w_i es variante conforme varía la potencia. Por último si $\lambda < 0$, entonces el factor $(1 - \lambda)$ es mayor a uno, lo cual

significa que el peso w_i decae exponencialmente conforme aumenta la potencia. Para este último sucede lo contrario, a mayor potencia menor peso. Analizando el comportamiento del factor λ se garantiza que para valores entre 0 y 1 el w_i se comporta exponencialmente creciente, es decir, a mayor potencia mayor peso y a menor potencia menor peso. El peso w_i de cada receptor R_i es calculado mediante la siguiente expresión:

$$w_i = (1 - \lambda)^{(v_{max} - v_i)} \quad (2.43)$$

El REWL centroide es calculado mediante la expresión (2.44):

$$\hat{p} = \frac{\sum_{i=1}^n (1 - \lambda)^{(v_{max} - v_i)} p_i}{\sum_{i=1}^n (1 - \lambda)^{(v_{max} - v_i)}} \quad (2.44)$$

En la figura 2.9, se muestra un ejemplo donde los pesos varían con respecto al RSS, en ésta se comparan los pesos relativos asignados a un conjunto de receptores con un rango de RSS de $v^\Delta = 15dBm$ teniendo en cuenta las asignaciones de pesos a RWL y REWL, asumiendo tres diferentes valores del factor de ponderación λ . Como se muestra en la Figura 2.9, la curva de ponderación obtenida del método RWL muestra un comportamiento lineal, lo cual se deduce de la expresión (2.41); las curvas de ponderación obtenidas del método REWL para diferentes valores del factor de ponderación λ , muestran un comportamiento exponencial, lo cual se deduce de la expresión (2.43), donde a menor valor de λ , la curva de ponderación tiende a variar en menor magnitud, lo cual se muestra para $\lambda = 0.1$.

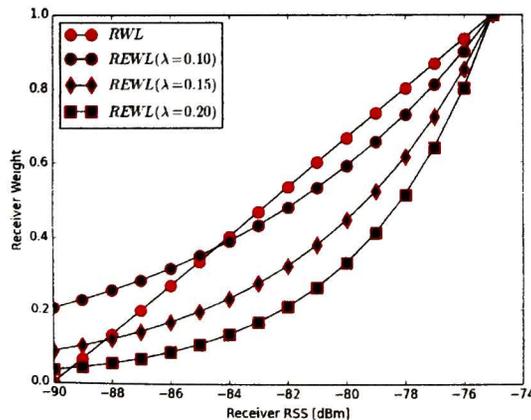


Fig. 2.9: Example of Relative Span Weights.

2.5.1.4 Algoritmo de Centroide con Sistemas Difusos

Es un método para encontrar los pesos de los nodos de referencia dados. El sistema de lógica difusa FLS (Fuzzy Logic System) es un método que mapea una entrada a un salida usando fuzzy logic. Este mecanismo consiste en un fuzificador, un mecanismo de inferencia y un defuzificador. El fuzificador, es el bloque encargado de asignar valores lingüísticos a la variable de entrada, por ejemplo: {bajo, medio, alto}. La salida de este bloque son funciones de membresía, El mecanismo de inferencia o sistema de inferencia difusa FIS (Fuzzy Inference System) es el encargado de tomar las funciones de membresía y generar la salida difusa mediante un conjunto de reglas o condiciones *if-then* y el defuzificador realiza la conversión del conjunto difuso a un valor crisp.

No existe un procedimiento sistemático para seleccionar una buena estrategia de defuzificación. La selección toma en consideración las propiedades de la aplicación en cada caso. Existen diferentes métodos. También existen diversos tipos de sistema difusos como Mamdani y Sugeno. La diferencia entre estos dos modelos son las funciones de pertenencia de salida, Mamdani utiliza funciones triangulares, trapezoidales, y Sugeno utiliza variables constantes o lineales.

Una regla difusa, puede ser escrita mediante la siguiente notación:

$$\text{Rule } i: \text{IF } x_1 \text{ is } A_1^i \text{ and } x_2 \text{ is } A_2^i \dots x_N \text{ is } A_N^i \text{ THEN } y \text{ is } y^i$$

donde $i = 1, 2, 3 \dots M$, denota la implicación i y M es el número de reglas, x_j $\{j = 1, 2, 3 \dots N\}$ son las variables de entrada del sistema difuso FLS, y_i es un singleton y A_j^i es la función de pertenencia difusa.

La salida del sistema difuso, para una entrada $X = \{x_1, x_2, \dots, x_N\}$ puede ser expresada por:

$$y = \frac{\sum_{i=1}^n \alpha_i y_i}{\sum_{i=1}^n \alpha_i} \quad (2.45)$$

donde α_i implica el valor de verdad de la premisa de la implicación i - *th*, y es calculado como:

$$\alpha_i = \prod_{j=1}^N A_j^i(x_j) \quad (2.46)$$

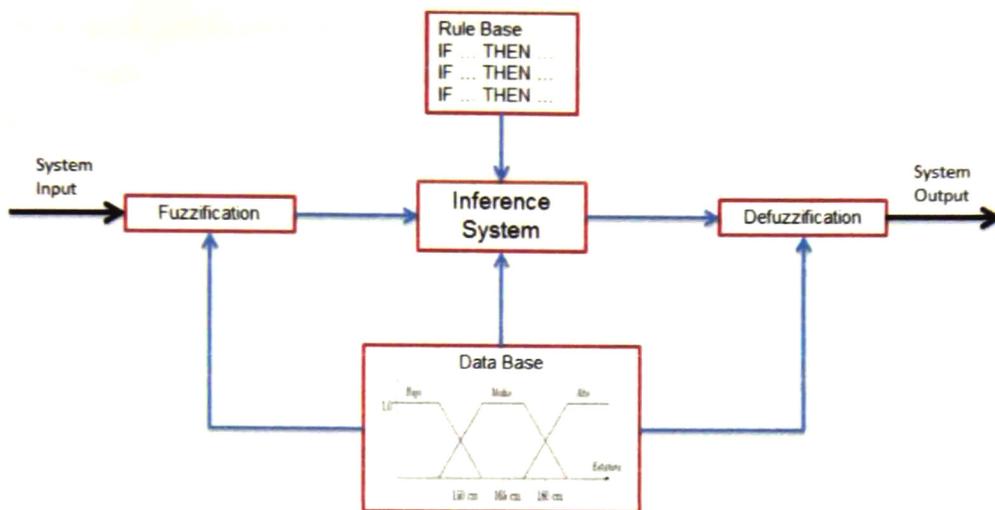


Fig. 2.10: Esquema FLS.

En este método, el cual fue aplicado para WSN, se utiliza el RSS como parámetro de entrada, el cual toma valores desde el intervalo $[RSS_{min}, 0]$ y el nivel de peso como variable de salida.

Tabla I. Reglas de lógica difusa para cada nivel de ponderación.

Rule	<i>IF: RSS is</i>	<i>THEN: Weight is</i>
Rule 1	<i>Very low</i>	<i>Very low</i>
Rule 2	<i>low</i>	<i>low</i>
Rule 3	<i>medium</i>	<i>medium</i>
Rule 4	<i>high</i>	<i>high</i>
Rule 5	<i>Very high</i>	<i>Very high</i>

En la Tabla I se muestran las reglas difusas asignadas para cada nivel de peso [21, 22]; como es probable que en algunos casos el nodo sensor reciba niveles bajos de potencia, entonces el nivel de peso debe ser bajo, en caso contrario el nivel de peso es alto.

Mamdani Fuzzy Inference System.

Este método fue modelado utilizando Mamdani fuzzy inference system [23, 24], y la variable de entrada (RSS) y la variable de salida (weight) fueron descompuestas en cinco

funciones de pertenencia trapezoidales simétricas: *very low*, *low*, *medium*, *high*, *very high*. En la Figura 2.11 se muestra las funciones de membresía de entrada y salida descompuestas en sus valores lingüísticos respectivamente.

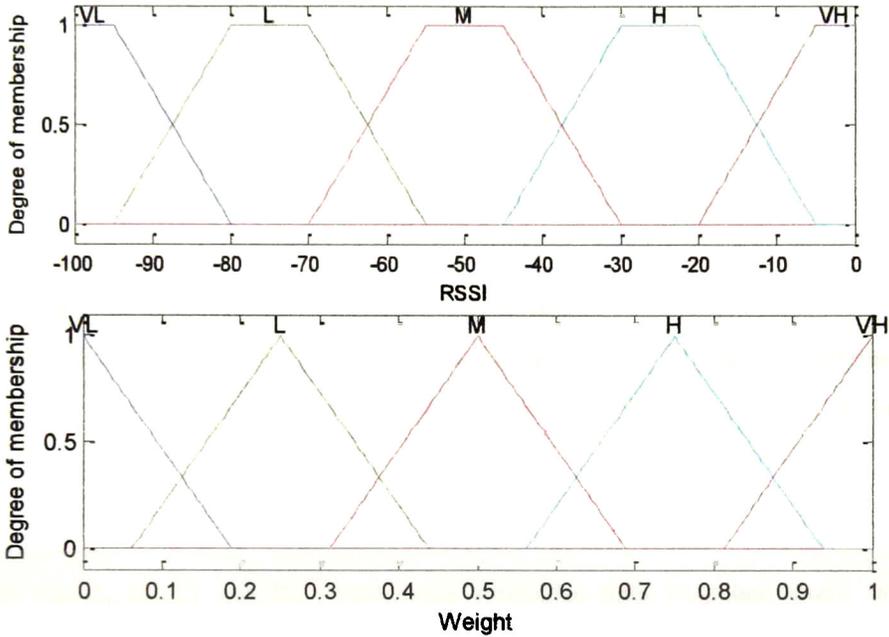


Fig. 2.11: Fuzzy Membership Function of RSSI and Weight Mamdani FIS.

Sugeno Fuzzy Inference System.

En este método el sistema lógico difuso fue modelado con Sugeno fuzzy inference [25, 26]; el parámetro de entrada *RSS* fue descompuesto en cinco funciones de pertenencia trapezoidales, nombradas {VL, L, M, H y VH}, y el parámetro de salida *Weight* fue descompuesto en cinco funciones de pertenencia lineales. En la Figura 2.12 se observan las descomposiciones para el parámetro de entrada (RSS). Finalmente para estimar la localización del NOI, se utiliza el algoritmo de centroide con pesos. Entonces la localización de la posición del NOI se realiza mediante la expresión (2.39).

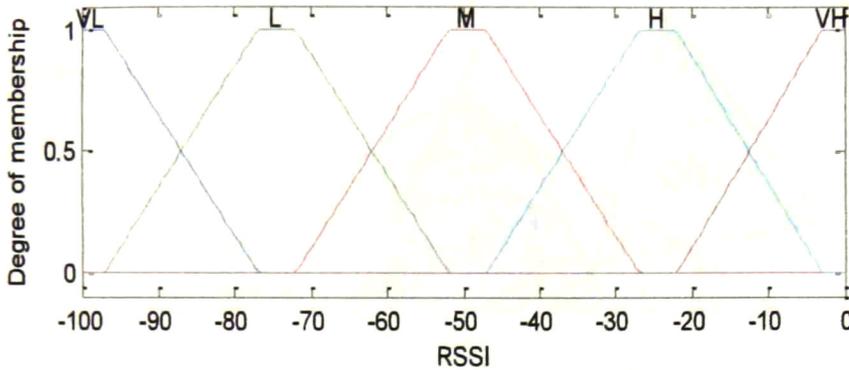


Figura 2.12: Fuzzy Membership function of RSS Sugeno FIS.

Combined Mamdani–Sugeno Fuzzy Inference System.

Para este método, la estimación de localización de un NOI, se realiza combinando los métodos de Mamdani [23] y Sugeno [25], entonces la localización del NOI resulta de la expresión (2.47).

$$\hat{p} = \frac{(x_{mam} + x_{sug}, y_{mam} + y_{sug})}{2} \quad (2.47)$$

Donde (x_{mam}, y_{mam}) son las coordenadas estimadas para Mamdani fuzzy inference system, (x_{sug}, y_{sug}) son las coordenadas estimadas para Sugeno fuzzy inference system.

2.5.2 Esquema de Localización APIT

Este método emplea un modelo basado en áreas, los nodos de referencia son divididos en regiones triangulares, cada nodo de interés puede estar dentro o fuera de estas regiones. En la Figura 2.13 se muestra el esquema de Localización APIT.

Este algoritmo puede ser ejecutado en cuatro pasos:

1. Intercambiar Balizas.
2. PIT test.
3. Agregación de APIT.
4. Calcular Centroide.

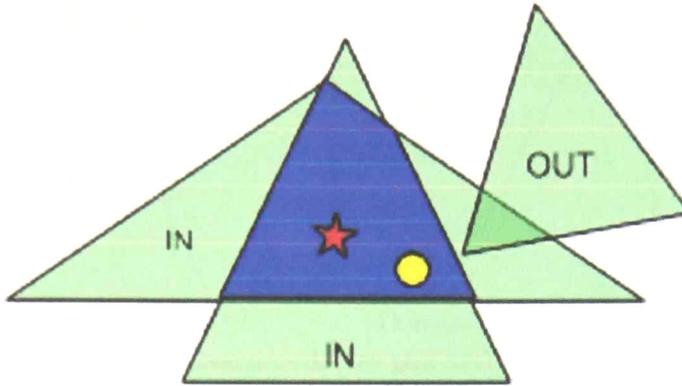


Fig. 2.13: Esquema de Localización APIT.

El pseudocódigo de este método es el siguiente:

- N nodos de referencia forman $\binom{N}{3}$ triángulos.
- Para cada Triángulo $T_i \in \binom{N}{3}$ {
 $Inside\ Set \leftarrow Point-In-Triangle-Test (T_i)$ }
- Posición = COG ($\cap T_i \in Inside\ Set$);

P.IT Test Aproximado: Si ningún vecino de M es más cercano o lejano a los nodos de referencia A, B y C simultáneamente, M asume que está dentro del triángulo ΔABC . En otro caso, M asume que se encuentra fuera del triángulo.

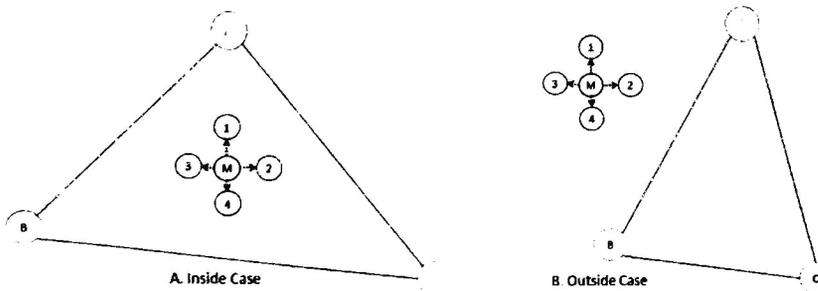


Fig. 2.14: Test PIT. a) Caso de adentro. b) Caso de afuera.

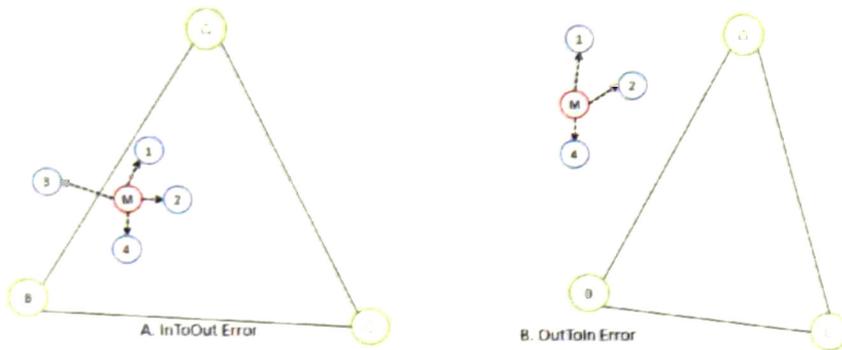


Fig. 2.15: Escenarios de error para APIT test.

En la figura 2.14 (A) se observa que ningún vecino del nodo M, está más cerca o lejos a los nodos de referencia A, B y C que el nodo M, por lo tanto el nodo está dentro del triángulo. En la figura 2.14 (B) el nodo 3 está más lejos a los 3 nodos de referencia que el nodo M, por lo tanto el nodo M, está fuera del triángulo.

En la figura 2.15 (A) el nodo 3 está más lejos a los tres nodos de referencia que el nodo M, entonces según el test el nodo M está fuera del triángulo, pero en la figura 2.14 (A) se observa que está dentro, por lo tanto se genera un error en el test y en la figura 2.15 (B) ninguno de los nodos vecinos 1, 2 y 4 está más cerca o lejos a los nodos de referencia que el M, entonces según el test el nodo está dentro del triángulo y en la figura se muestra que fuera del triángulo, entonces hay un error en el test. Entonces, estos serían los posibles casos donde el test genere un error.

Agregación de APIT: en este método un grid formado por un conjunto de puntos es utilizado para representar el área máxima en donde es probable obtener la localización del nodo de interés. La resolución del grid puede ser alterada para mejorar la precisión del algoritmo, es decir a mayor resolución, la localización del centroide es más precisa.

En la Figura 2.16 se muestra un ejemplo de cómo encontrar el área de traslape de varios triángulos, donde se propone un grid imaginario, en el cual las casillas con mayor ponderación indican la región del área común del grupo de triángulos.

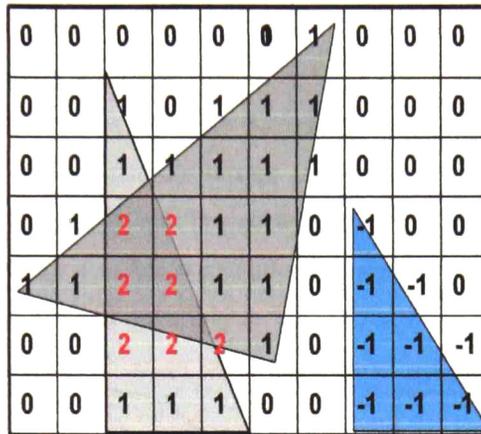


Fig. 2.16: Método SCAN para el cálculo del área de traslape.

El algoritmo propuesto para encontrar el área de traslape de un conjunto de triángulos es el siguiente:

Input: $T[m][6]$ {Arreglo de triángulos con vértices X y Y. $m = \text{número de triángulos}$ }.

Output: $OverlapArea[n][n]$ {Matriz que contiene los puntos del área de traslape}.

- 1: Encontrar los límites en X y Y del conjunto de triángulos, es decir $\{x_{min}, x_{max}, y_{min}, y_{max}\}$.
- 2: Proponer un grid "G" de puntos dentro de las coordenadas que limitan el conjunto de triángulos de tamaño $n \times n$.
- 3: Inicializar $OverlapArea = 0$;
- 4: Para $i = 1:m$ hacer
 - 5: Para $j = 1:n$ hacer
 - 6: Para $k = 1:n$ hacer
 - 7: Si $G[j, k]$ está dentro de $T[m, :]$ entonces
 - 8: Auxiliar[j, k, i] = 1;
 - 9: Si no
 - 10: Auxiliar[j, k, i] = 0;
 - 11: Fin Si
 - 12: Fin Para
 - 13: Fin Para
 - 14: $OverlapArea = Auxiliar[:, :, i] + OverlapArea$;
 - 14: Fin Para
 - 15: Regresar $OverlapArea$;

2.5.3 Intersección Rectangular

Este esquema está basado en la conectividad de nodos, es decir si un nodo está conectado a dos nodos de referencia, entonces se asume que está dentro del cuadrado de cada nodo de referencia. La ventaja de este método es que la intersección de los cuadrados es una operación matemática más simple que la intersección de círculos. Esto se debe a que la intersección de dos cuadrados es un rectángulo y la de dos círculos es una figura más compleja, como se muestra en la Figura 2.17.

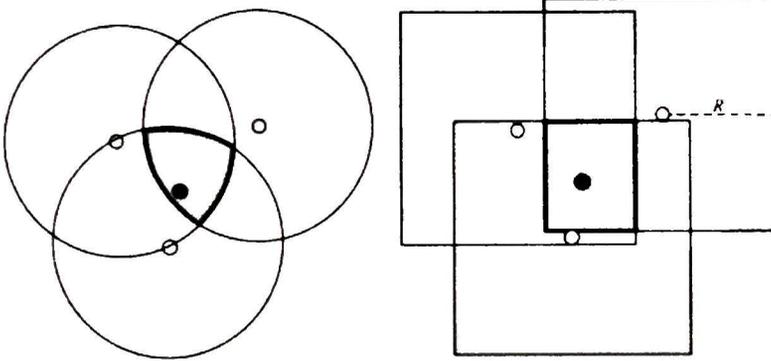


Fig. 2.17: Intersección Rectangular. a) Intersección Circular. b) Intersección Rectangular.

Cada rectángulo se puede representar por sus esquinas superior izquierda e inferior derecha. Si todos los nodos vecinos tienen sus coordenadas en los puntos x_i, y_i se definen las esquinas del rectángulo de intersección como:

$$X_{SI}, Y_{SI} = \left[\max_{i \in 1 \dots N} x_i - R, \max_{i \in 1 \dots N} y_i - R \right]$$

$$X_{ID}, Y_{ID} = \left[\min_{i \in 1 \dots N} x_i + R, \min_{i \in 1 \dots N} y_i + R \right]$$

donde R es el radio de cobertura de los nodos vecinos y N es el número de nodos vecinos. La posición del nodo de interés será el centroide del rectángulo de intersección.

2.5.4 K Vecino más Cercano

Este método está dividido en dos etapas: offline y online [27]. En la primera etapa se mide el valor de RSS de cada fingerprint (puntos de localización donde miden varios valores de

RSS de cada nodo de referencia), y se filtran aquellos valores fuera de rango y se calcula un valor de RSS promedio para cada fingerprint y este valor se almacena en una base de datos. Durante la etapa online los fingerprints de los K nodos, los cuales tienen los valores de RSS más cercanos a los valores medidos por el NOI, son encontrados. Entonces, esos fingerprints son los K vecinos más cercanos. Asumiendo RSS_{ij} , como el valor medido del fingerprint j a la referencia i , donde $j = [1, 2, \dots, m]$, m es el número de fingerprints y RSS_{Ti} es el valor medido del NOI a la referencia i . Entonces los vecinos más cercanos son buscados mediante la distancia euclidiana entre los valores de RSS, tal como se muestra en (2.48).

$$d_j = \sqrt{\sum_{i=0}^n (RSS_{Ti} - RSS_{ij})^2} \quad (2.48)$$

Después de calculada la distancia con respecto al conjunto total de muestras, un conjunto de K muestras de datos de la base de datos se selecciona en función de la distancia más pequeña. Finalmente, promediando estas coordenadas de K vecinos, la ubicación (x, y) online de la señal recogida puede ser estimada por:

$$\hat{p} = \frac{1}{K} \sum_{k=1}^K (x_k, y_k) \quad (2.49)$$

donde (x_k, y_k) son las coordenadas del vecino k más cercano, K es el número de vecinos más cercanos al NOI.

2.5.4.1 K-Vecinos más Cercanos con Lógica Difusa

En el método de K vecinos más cercanos KNN (K Nearest Neighbor) se utiliza la distancia euclidiana que determinan los K vecinos más cercanos, como parámetro de entrada para calcular los pesos mediante sistemas difusos.

Sistema de Inferencia Difusa

En este método son preferidas las inferencias tipo Sugeno que Mamdani [27], cinco constantes son propuestas a utilizar como salida de la función de pertenencia con valores $\{0, 0.25, 0.5, 0.75, 1\}$, las cuales tienen valores lingüísticos de $\{\text{muy pequeño, pequeño,}$

medio, grande y muy grande). En las Figuras 2.18 y 2.19 se observan dichas descomposiciones.

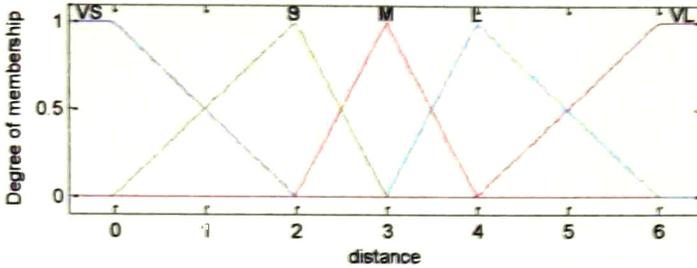


Fig. 2.18: Función de pertenencia de entrada Fuzzy-KNN.

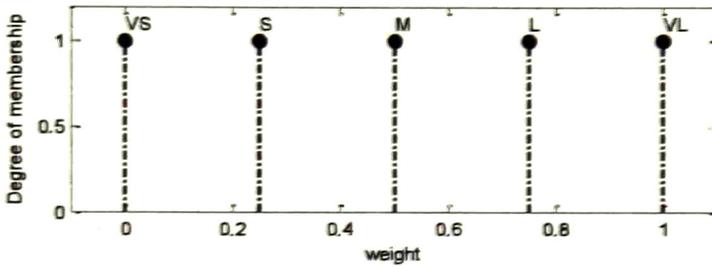


Fig. 2.19: Función de pertenencia de salida Fuzzy-KNN.

Hay cinco reglas propuestas para este método.

- Si (la distancia es muy cercana), entonces (el peso es muy grande).
- Si (la distancia es cercana), entonces (el peso es grande).
- Si (la distancia es media), entonces (el peso es medio).
- Si (la distancia es lejana), entonces (el peso es pequeño).
- Si (la distancia es muy lejana), entonces (el peso es muy pequeño).

El peso es calculado por FIS y las coordenadas de los K vecinos más cercanos son utilizadas para calcular la posición del NOI. Las coordenadas del NOI son calculadas mediante (2.50).

$$\hat{p} = \frac{\sum_{k=1}^K (x_k, y_k) w_k}{\sum_{k=1}^K w_k} \quad (2.50)$$

donde (x_k, y_k) son las coordenadas del vecino k más cercano, K es el número de vecinos más cercanos al NOI, w_k es el peso asignado a cada vecino k .

Capítulo 3

Evaluación de Desempeño de las Técnicas de Localización

En este capítulo se evalúa las métricas de desempeño de los algoritmos de localización. Este capítulo está estructurado de la siguiente manera: primero se describen las métricas de desempeño de los algoritmos de localización, así mismo, los escenarios de localización y por último se presentan los resultados obtenidos.

Los algoritmos evaluados están basados en RSS para estimar la posición del nodo de interés, ya que los otros métodos de estimación de distancia, involucran hardware más costoso en el caso de AoA, el método ToA involucra que no siempre haya sincronización perfecta entre todos los nodos y el método TDoA está limitado en cuanto a cobertura [9], por lo general de pocos metros. La desventaja del método RSS, es que a medida que aumenta la distancia de transmisión, tiende a ser impreciso en la estimación de la distancia. Dentro los algoritmos evaluados, se consideran los algoritmos libres de distancia, como: CL, WCL, RWL, REWL, centroide con FIS y KNN, y de los algoritmos basados en distancia se evalúa el algoritmo Hiperbólico, hiperbólico ponderado, circular y multilateración WLS.

En este capítulo, se evalúan dos escenarios, un primer escenario en un área de cobertura extensa y un segundo escenario en un área pequeña. Las métricas de desempeño a evaluar son exactitud en cuanto a MSE, precisión en cuanto a desviación estándar SD (Standard Deviation) y complejidad computacional. Los resultados de las métricas de desempeño, fueron simulados en Matlab versión 2011.

3.1 Parámetros del Desempeño

Exactitud. Este parámetro se define como el valor esperado de los errores de localización. En la ecuación 3.1 se muestra el cálculo para obtener esta métrica de desempeño.

$$MSE = E(\sqrt{(x - x_k)^2 + (y - y_k)^2}) \quad (3.1)$$

$$location\ error = \sqrt{(x - x_k)^2 + (y - y_k)^2} \quad (3.2)$$

donde:

(x, y) = real position node.

(x_k, y_k) = estimated position node.

Precisión. Considera la distribución de las distancias de errores mientras que la exactitud considera el valor medio de los errores. Cuando dos técnicas son comparadas, la técnica con distancias de error concentradas sobre valores pequeños es la preferida.

Complejidad Computacional. Este parámetro se refiere al número de operaciones matemáticas que realiza un algoritmo durante su ejecución. En este trabajo, este parámetro se mide mediante el tiempo promedio de cómputo que tarda un algoritmo en estimar la posición de un nodo para cada densidad de nodos, también se evalúa la complejidad del algoritmo considerando su orden de complejidad.

3.2 Descripción del Primer Escenario de Evaluación

El primer escenario en el cual se analizaron las técnicas de localización está representado por un área de 1000m x 1000m, tal como se muestra en la Figura 3.1. Se evaluaron 10,000 iteraciones, para cada una de las cuales se genera un nodo aleatorio, que representa el nodo a localizar. Para toda densidad de nodos $\rho \in \{1,2,5\}$, estos son distribuidos aleatoriamente sobre todo el grid de simulación. Para cada una de las densidades de los nodos, se varía el nivel de ruido (desviación estándar) donde $\sigma \in \{0.5,1,2,3,4,5,6,7,8\}$. Para cada nodo de interés, se calcula el valor de RSS con el modelo de propagación log-normal Shadowing ec. (2.3).

Los parámetros del modelo de propagación de la señal fueron propuestos de: $n = 4$, (exponente de pérdidas en el espacio libre) y $P_o(dB) = -12dBm$ la cual fue calculada considerando un sensor con un rango de transmisión de 100m en exteriores, una

sensibilidad de -92dBm y una potencia de transmisión de 1mW (0dBm). d_{ij} es la distancia entre el nodo de interés i y el nodo de referencia adyacente j . X_σ es una variable aleatoria con ruido AWGN.

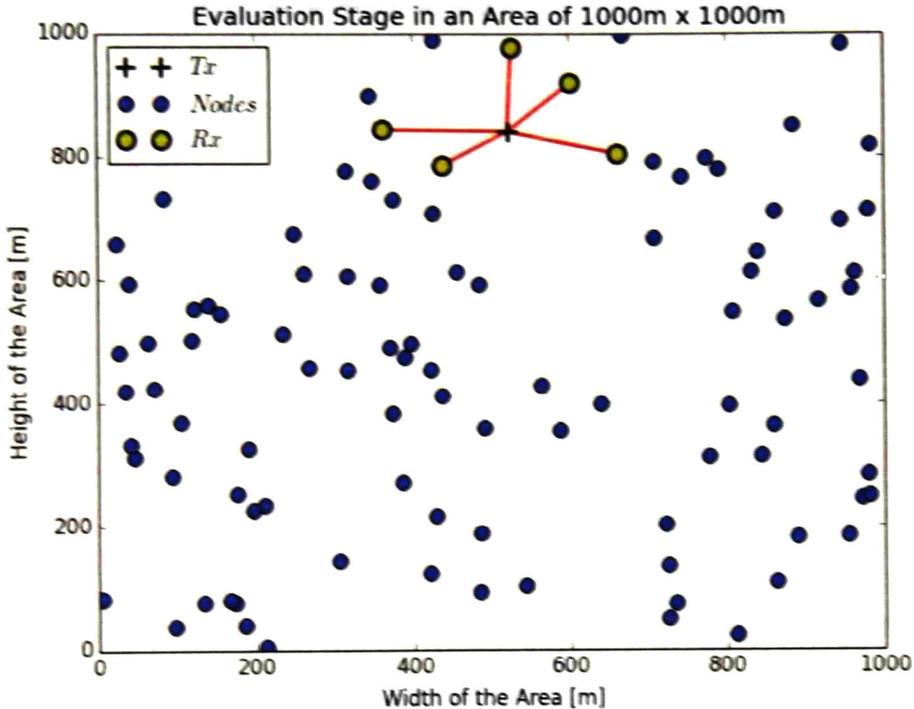


Fig. 3.1: Distribución de nodos.

3.3 Análisis del Desempeño de Algoritmos libres de distancia bajo escenario 1

Análisis de MSE y SD

En la Figura 3.2, se observa el MSE y la desviación estándar obtenida para el algoritmo de localización para una densidad de nodos de 1 (nodo/100m x 100m). En cuanto a MSE, se observa que el algoritmo de multilateración WLS presenta mejor desempeño que los demás algoritmos por debajo de $\sigma = 5\text{ dBm}$ de nivel de ruido; para niveles de ruido mayores a este valor, su MSE es mayor que los algoritmos REWL, RWL y WCL. En cuanto a precisión este algoritmo tiene un mejor comportamiento que los demás algoritmos para niveles de ruido por debajo de $\sigma = 2\text{ dBm}$, ya que para niveles de ruido bajo, el nodo de

interés es más probable a ser localizado dentro del área de cobertura de los nodos de referencia, lo cual propicia a obtener una buena precisión, en cambio para niveles altos de ruido, el nodo de interés es susceptible a ser localizado fuera del área de cobertura de los nodos ancla, lo cual implica una mala precisión.

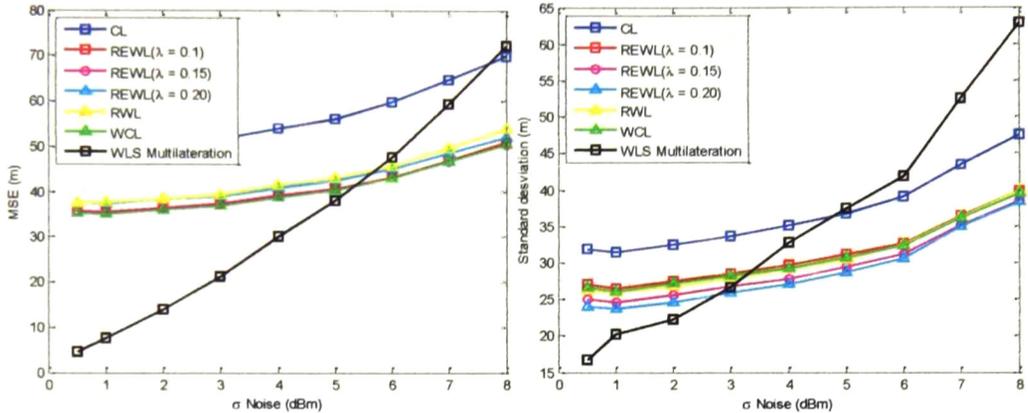


Fig. 3.2: MSE and SD vs. (σ Noise) for node density = 1.

En la Figura 3.3, se observa el MSE y la desviación estándar obtenida para algoritmo de localización para una densidad de nodos de 2 (nodos/100m x 100m). Analizando la Figura 3.3, en cuanto a desempeño, todos los algoritmos presentaron menor MSE y mayor precisión, debido a que aumenta la densidad de nodos. Para esta densidad de nodos, se observa que el algoritmo de multilateración WLS presenta mejor precisión hasta niveles de ruido con $\sigma = 4 \text{ dBm}$, ya que el aumento de densidad de nodos, implica un aumento en la cantidad de información de las distancias conocidas entre el NOI y los nodos ancla, por lo que es más probable a estimar una posición más cercana a la posición real del NOI; por ello la curva de desviación obtenida para este algoritmo de localización, se observa que su comportamiento es más regular, comparado en la Figura 3.2.

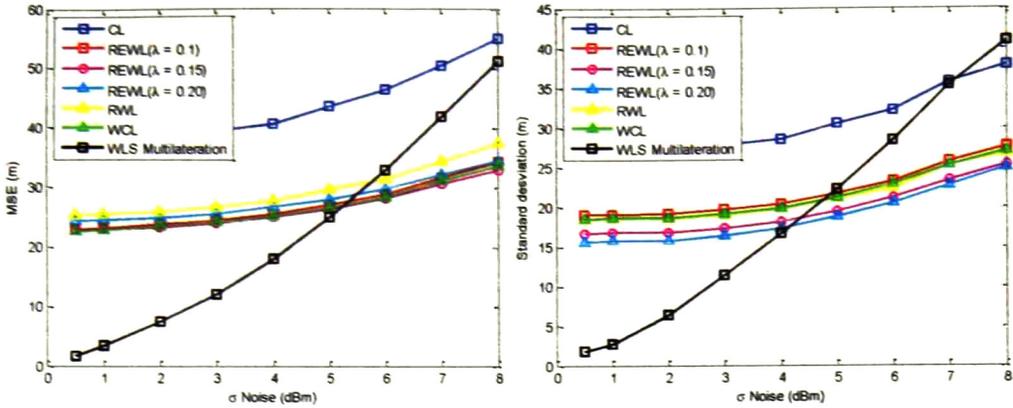


Fig. 3.3: MSE and SD vs. (σ Noise) for node density = 2.

Analizando la Figura 3.4, se observa que para una densidad de nodos de 5 (nodos/100m x 100m) todos los algoritmos presentan menor MSE y mayor precisión. Entonces, se puede concluir que un aumento en la densidad de nodos para todo nivel de ruido, implica mejor desempeño en cuanto a MSE, ya que éste tiende a bajar y repercute en una mejor precisión, ya que ésta tiende a aumentar.

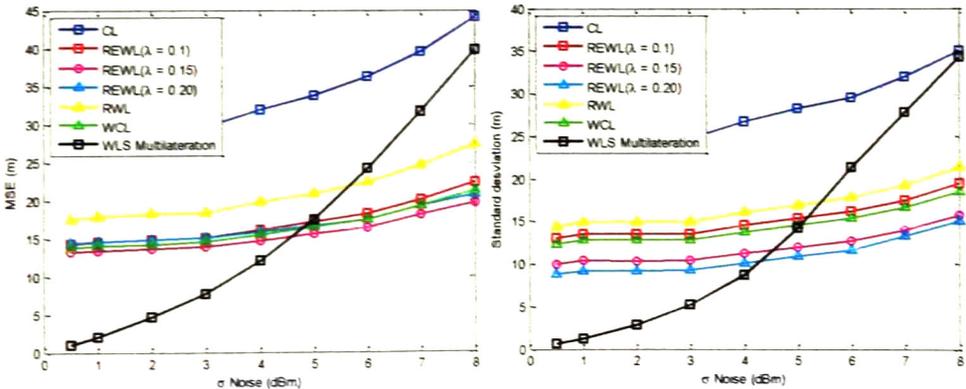


Fig. 3.4: MSE and SD vs. (σ Noise) for node density = 5.

En la Figura 3.5, se observa cómo descende el MSE para cada algoritmo de localización conforme aumenta la densidad de nodos en la red.

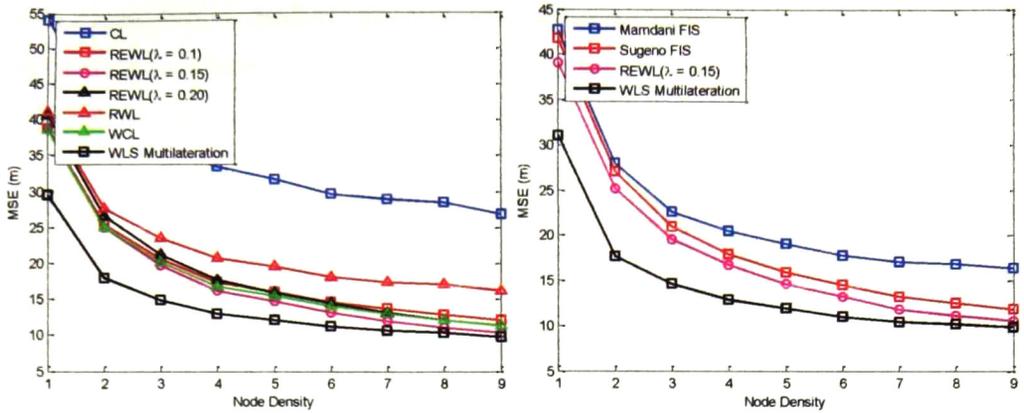


Fig. 3.5: MSE vs. Node density for (σ Noise = 4 dBm).

En la Figura 3.5, se observa que el algoritmo de multilateración WLS, muestra mejor desempeño en cuanto a MSE conforme aumenta la densidad de nodos en la red. En cuanto a sistemas difusos, se observa que Sugeno FIS se desempeña mejor que Mamdani FIS, por lo que es más adecuado usar Sugeno FIS para estimar la posición de un nodo.

En la figura 3.6, se observa como varía la desviación estándar conforme aumenta la densidad de nodos en la red. Analizando la desviación estándar, se observa que el algoritmo de multilateración WLS, presenta menor desviación estándar que los demás algoritmos analizados, por lo que este método es más preciso.

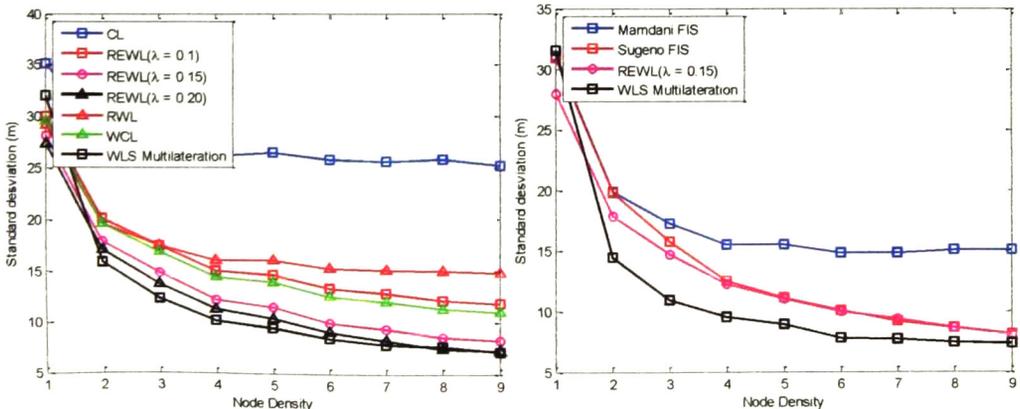


Fig. 3.6: SD vs. Node density for (σ Noise = 4 dBm).

En la Figura 3.7, se muestra que para una desviación estándar de ruido de 1 dBm, el algoritmo de multilateración WLS, mantiene un MSE muy bajo comparado con los demás algoritmos mostrados en la Figura 3.7.

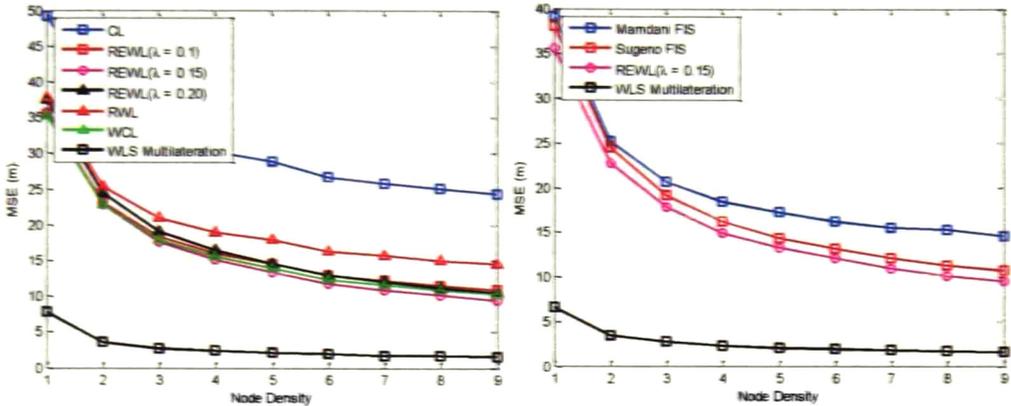


Fig. 3.7: MSE vs. Node density for (σ Noise = 1 dBm).

En la Figura 3.8, se muestra que para una desviación estándar de ruido de 1 dBm, el algoritmo de multilateración WLS, mantiene una precisión muy alta comparado con los demás algoritmos mostrados en dicha figura. Para una densidad de un (nodo/100m x 100m) la precisión es baja, ya que son pocos los nodos de referencia para estimar la localización del nodo de interés.

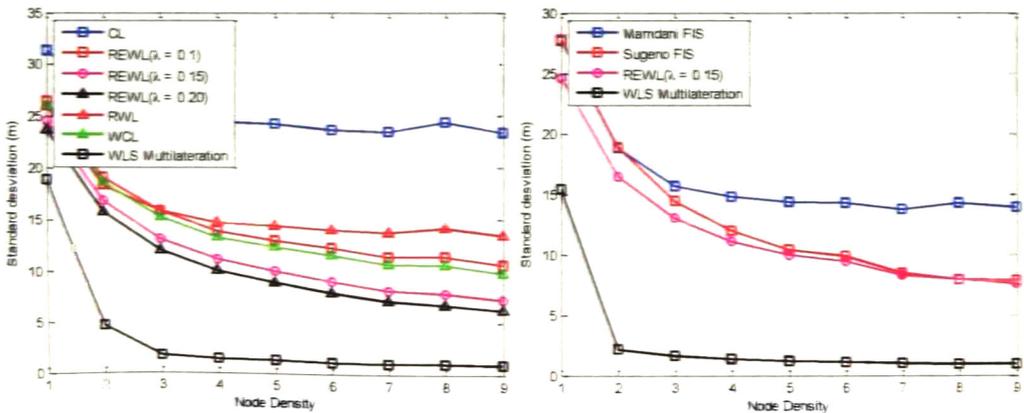


Fig. 3.8: SD vs. Node density for (σ Noise = 1 dBm).

A continuación se muestran unas imágenes, donde se observa la precisión de todos los algoritmos analizados para este escenario de evaluación, conforme varía el error de localización mediante la función de distribución acumulada CDF (Cumulative Distribution Function).

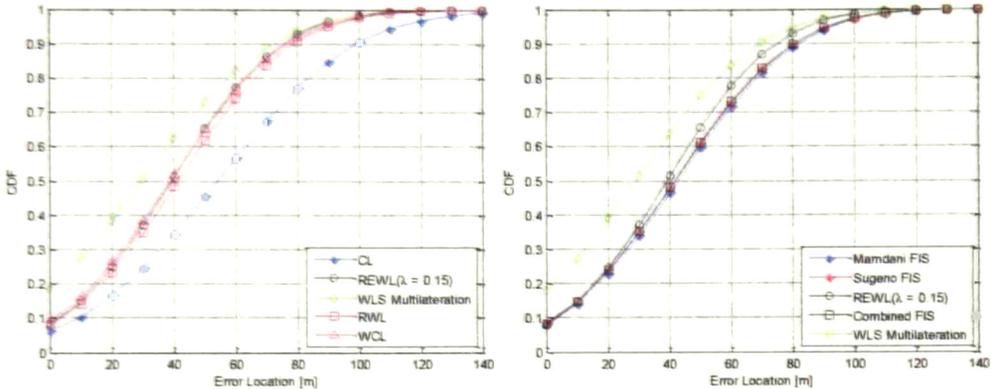


Fig. 3.9: CDF vs Error Location for (σ Noise = 4 dBm and Node Density = 1).

En la Figura 3.9 se observa que el algoritmo de multilateración WLS presenta mejor precisión que los demás algoritmos analizados. Para un 90% de los casos, presenta error de localización menor a 70m, para el 50% de los casos presenta error de localización menor a 30m.

En la figura 3.10 se observa que la precisión del algoritmo de multilateración WLS es considerablemente mejor que la de los demás algoritmos analizados. Para un 90% de los casos, presenta error de localización menor a 30m, para el 50% de los casos presenta error de localización menor a 10m.

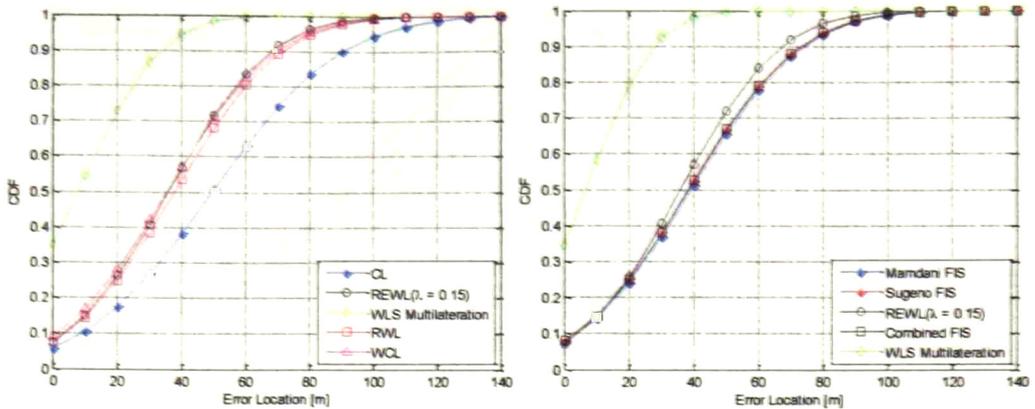


Fig. 3.10: CDF vs. Error Location for (σ Noise = 1 dBm and Node Density = 1).

3.4 Análisis del Desempeño de Algoritmos basados en distancia bajo escenario 1

Análisis de MSE y SD

En las figuras 3.11 y 3.12 se observa el desempeño de los algoritmos de localización para densidades de nodos de 1 y 2 (nodos/100m x 100m); comparando los algoritmos analizados, se observa que el algoritmo de multilateración WLS presenta mejor MSE y desviación estándar que los demás algoritmos analizados para ambas densidades de nodos. Para este análisis, sólo se tomó el algoritmo de multilateración WLS para ser comparado con otras técnicas, ya que este algoritmo mostró mejor desempeño que los algoritmos de centroide presentados anteriormente.

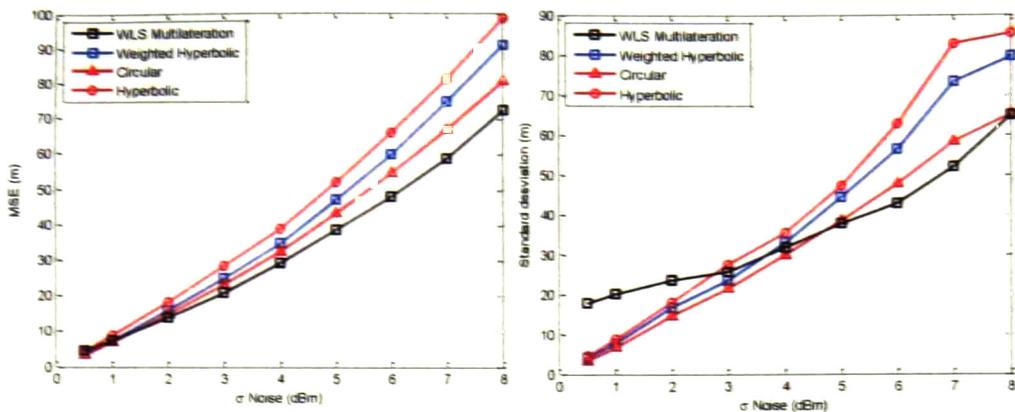


Fig. 3.11: MSE and SD vs. (σ Noise) for Node Density = 1.

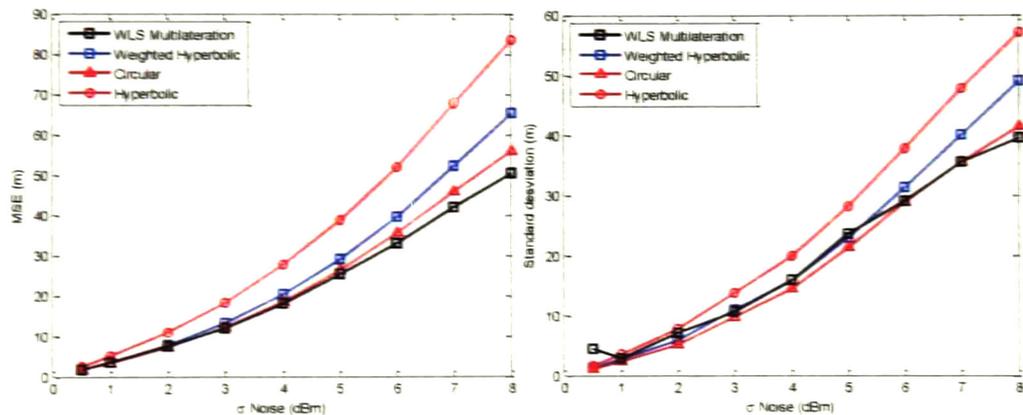


Fig. 3.12: MSE and SD vs. (σ Noise) for Node Density = 2.

En la Figura 3.13, se observa que el de mutilateración WLS y el algoritmo circular presentan el mismo desempeño de MSE, aunque en desviación estándar, el algoritmo circular se desempeña mejor.

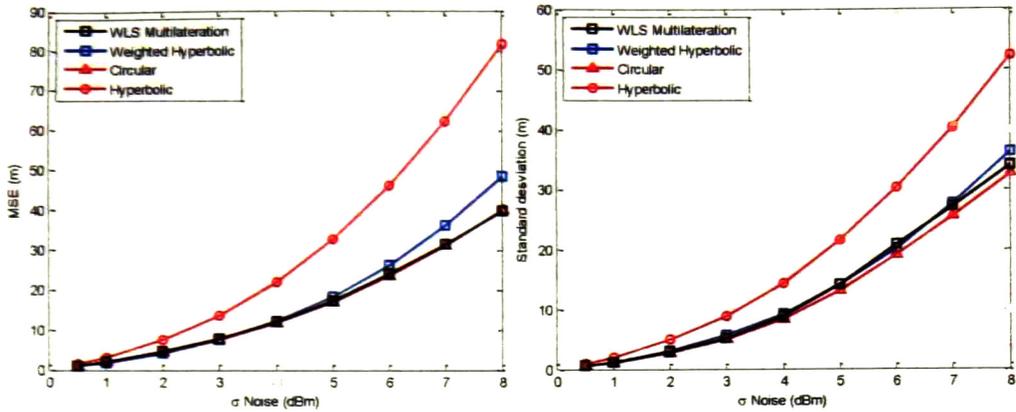


Fig. 3.13: MSE and SD vs. (σ Noise) for Node Density = 5.

Esta mejora del algoritmo circular comparado en los esquemas anteriores, se debe a la mejora del algoritmo Weighted Hyperbolic, ya que la posición calculada por este algoritmo es la posición inicialmente estimada por el algoritmo circular. El algoritmo circular, al igual que el algoritmo de multilateración WLS es iterativo, por lo que el usar la posición calculada por el algoritmo Weighted Hyperbolic, propicia que el algoritmo circular calcule una posición con menor error durante el número de iteraciones para converger a la posición estimada.

A continuación en las Figuras 3.14 y 3.15 se muestran las gráficas de desempeño de MSE y SD variando la densidad de nodos en la red para desviaciones estándar de ruido gaussiano de 1 dBm y 4 dBm.

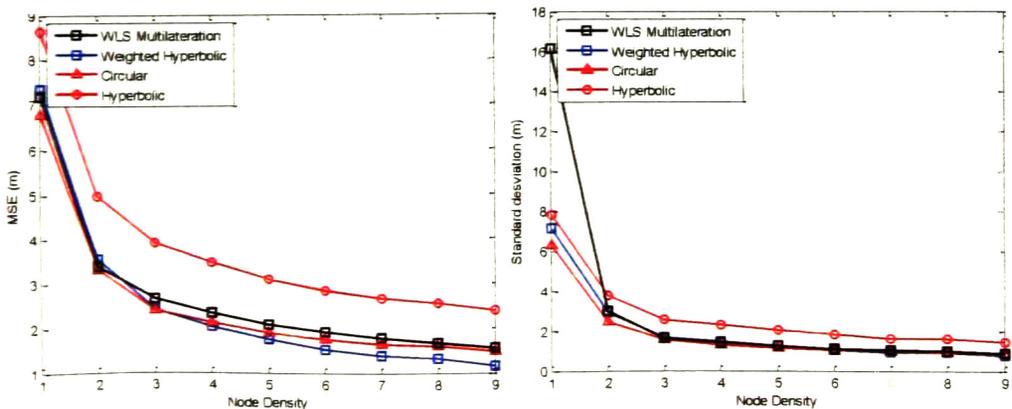


Fig. 3.14: MSE and SD vs. Node density for (σ Noise = 1 dBm).

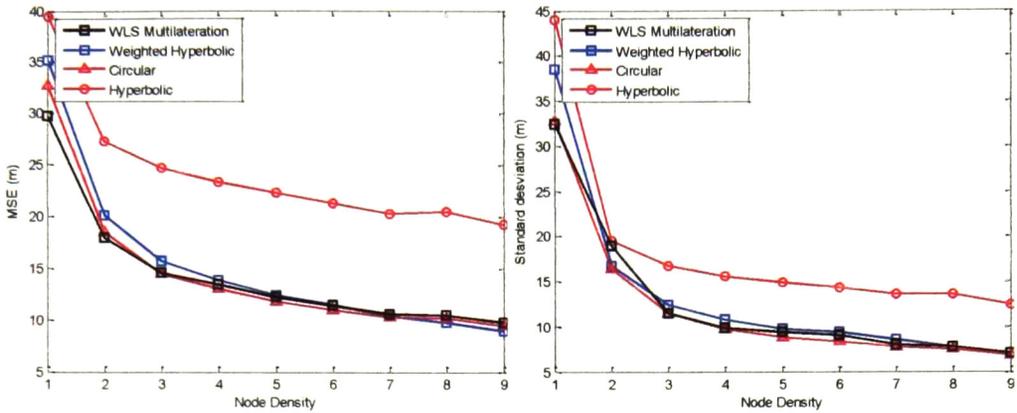


Fig. 3.15: MSE and SD vs. Node density for (σ Noise = 4 dBm).

En la Figura 3.14 se observa que el algoritmo de posicionamiento hiperbólico ponderado presenta un mejor desempeño de MSE para densidades superiores a 5 nodos por 100m x 100m que los demás algoritmos mostrados en la figura y, además, los algoritmos mantienen cierta robustez para densidades superiores a 6 nodos por 100m x 100m. En la Figura 3.15 el desempeño de los algoritmos de localización es similar a partir de densidades superiores a 5 nodos por 100m x 100m.

3.5 Descripción del Segundo Escenario de Evaluación

La figura 3.16 muestra el escenario de evaluación propuesto, en el que se evaluaron los algoritmos de localización de Centroide, Fuzzy Centroide, Multilateración WLS y Vecino más Cercano, para un área de 10m x 10m con 4 nodos de referencia.

La fuerza de la señal recibida fue calculada mediante el modelo log-normal shadowing. Donde, $n = 4$, (exponente de pérdidas en el espacio libre) y $P_o(dB) = -40dBm$ como potencia mínima a una distancia de 1m, la cual fue calculada considerando un sensor con un rango de transmisión de 20m en interiores, una sensibilidad de $-92dBm$ y una potencia de transmisión de 1mW (0dBm).

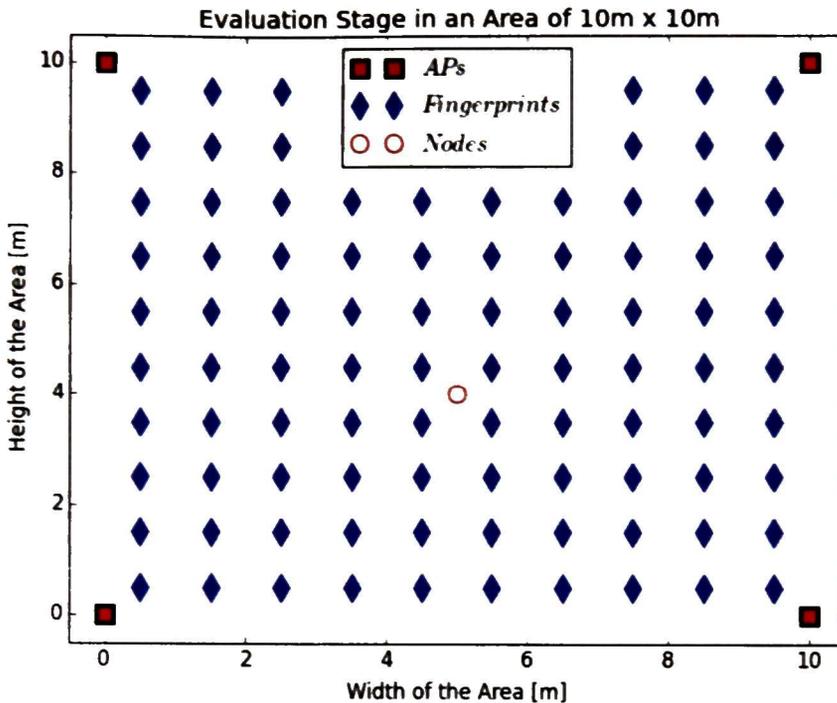


Fig. 3.16: Escenario de Aplicación propuesto.

3.6 Análisis del Desempeño de los Algoritmos Analizados bajo escenario 2

La Figura 3.17 presenta el desempeño en cuanto a MSE de los algoritmos de localización analizados para este escenario. Para niveles de ruido inferiores a $\sigma = 2 \text{ dBm}$, WLS Multilateration presenta menor MSE que todos los algoritmos presentados. Haciendo una comparación general de los algoritmos analizados, el algoritmo REWL ($\lambda = 0.1$) presenta menor MSE para niveles de ruido por encima de $\sigma = 3 \text{ dBm}$ que los demás algoritmos. Entonces para niveles de ruido inferiores a $\sigma = 2 \text{ dBm}$, multilateración WLS se desempeña mejor y para niveles superiores a $\sigma = 3 \text{ dBm}$, REWL ($\lambda = 0.1$) se desempeña mejor.

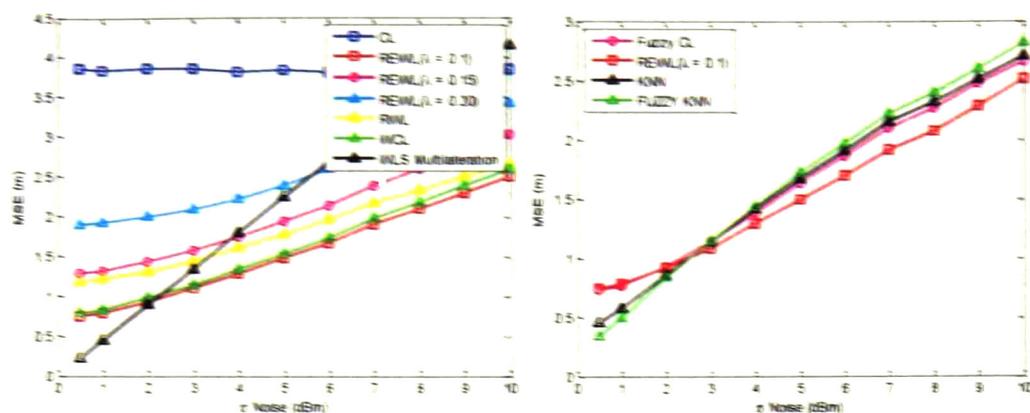


Fig. 3.17: MSE vs. (σ Noise).

Para niveles de ruido inferiores a $\sigma = 2 \text{ dBm}$, WLS Multilateration presenta mejor desempeño. Comparando KNN con Fuzzy KNN, se observa que el algoritmo Fuzzy KNN, presenta mejor desempeño para niveles de ruido inferiores a $\sigma = 3 \text{ dBm}$; en cambio para niveles de ruido superiores a este valor, se desempeña mejor KNN.

En la Figura 3.18 se muestra cómo varía la desviación estándar para los algoritmos analizados. Se observa que para niveles de ruido inferiores a $\sigma = 2 \text{ dBm}$, el algoritmo Fuzzy KNN es más preciso que el algoritmo KNN, en cambio para niveles superiores a este valor, KNN es más preciso. Para niveles altos de ruido, el algoritmo REWL ($\lambda = 0.1$) es más preciso que todos los algoritmos evaluados en este escenario, en cambio para niveles bajos de ruido, multilateración WLS es más preciso. En conclusión para condiciones donde el ruido sea muy bajo, multilateración WLS es la mejor opción para estimar la posición de un NOI, ya que resultó ser algoritmo con menor MSE y mejor precisión. Si se desea utilizar la información de los fingerprints para estimar la posición del NOI, considerando que el ruido gaussiano es bajo, es preferible utilizar Fuzzy KNN que KNN para estimar la posición del NOI.

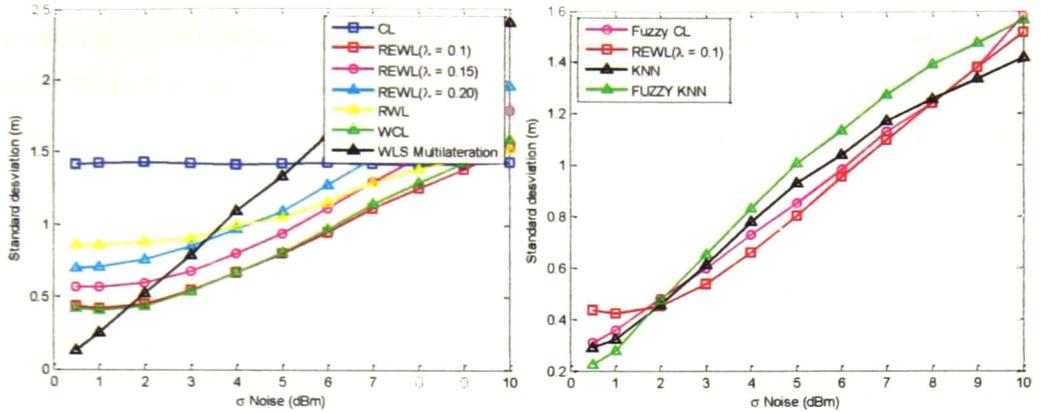


Fig. 3.18: SD vs. (σ Noise).

En la Figura 3.19 presenta las curvas de CDF para los algoritmos de localización, analizados en este escenario, para $\sigma = 1 \text{ dBm}$, en la que se observa que WLS Multilateration, presenta mejor precisión que todos los algoritmos analizados, para el 90% de los casos. El error de localización de este algoritmo es menor a 1m y en el 50% de los casos el error es menor a 0.5m.

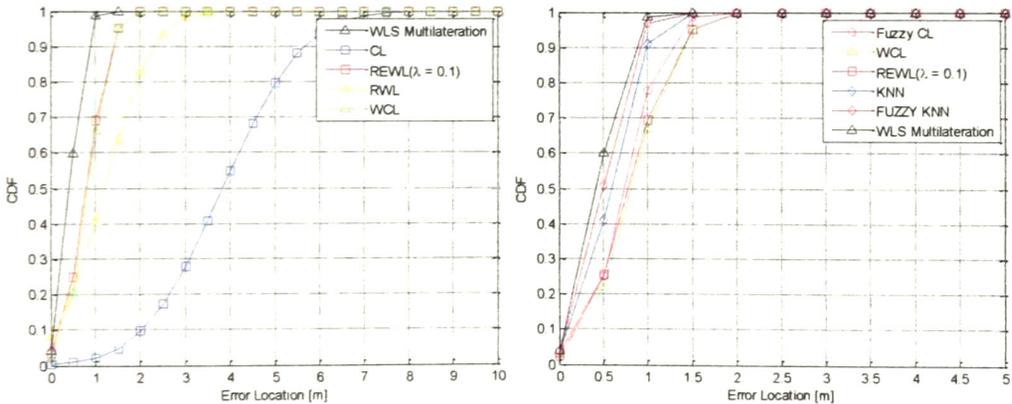


Fig. 3.19: CDF vs. Error Location for (σ Noise = 2).

3.7 Análisis de la Complejidad Computacional

La evaluación de la complejidad computacional de los algoritmos de localización basados en distancia y libres de distancia es muy importante, ya que esta métrica de desempeño determina el costo computacional del algoritmo tanto en hardware como en software o bien

la cantidad de recursos computacionales utilizados por el algoritmo para solucionar un problema. La complejidad computacional de los algoritmos de localización presentados en este trabajo fue evaluada desde un ordenador midiendo el promedio del tiempo de cómputo que tarda el algoritmo durante su ejecución y el número de operaciones elementales que realiza el algoritmo durante su ejecución.

Al emplear ambos métodos se llegan a diferentes resultados, pero el primer método empleado para medir la complejidad computacional tiene la desventaja de que los resultados obtenidos no son fiables, ya que el tiempo de procesamiento de cada algoritmo varía de dispositivo a dispositivo, por lo que no se tiene la certeza de obtener resultados con cierta precisión. En cambio el segundo método garantiza obtener el número de operaciones elementales por segundo de cada algoritmo de localización independientemente del dispositivo donde éste se ejecute.

3.7.1 Complejidad Computacional medida desde un ordenador

Para medir el tiempo de ejecución de cada algoritmo de localización se midió el tiempo de cómputo en una máquina con un procesador AMD A4-3300M APU con Radeon™ HD Graphics 1.90 GHz y una memoria RAM de 4,00 GB.

A continuación, se muestran unas gráficas del tiempo de cómputo promedio para cada densidad de nodos, donde $\rho \in \{1,2,3,4,5,6,7,8,9\}$. Estas evaluaciones del tiempo de procesamiento promedio, fueron realizadas durante un ciclo de 10,000 iteraciones para cada densidad de nodos. Los algoritmos evaluados son libres en distancia y basados en distancia.

La Figura 3.20 muestra el tiempo de procesamiento promedio para los algoritmos de centroide CL, WCL, RWL y REWL, donde se observa que REWL es el algoritmo más complejo.

En la Figura 3.21 se muestra una comparación del tiempo de procesamiento de los algoritmos de centroide Mamdani FIS y Sugeno FIS, donde el algoritmo Sugeno FIS es más eficiente computacionalmente que Mamdani FIS, sin embargo, estos algoritmos son

computacionalmente más costosos que los algoritmos de centroide CL, WCL, RWL y REWL.

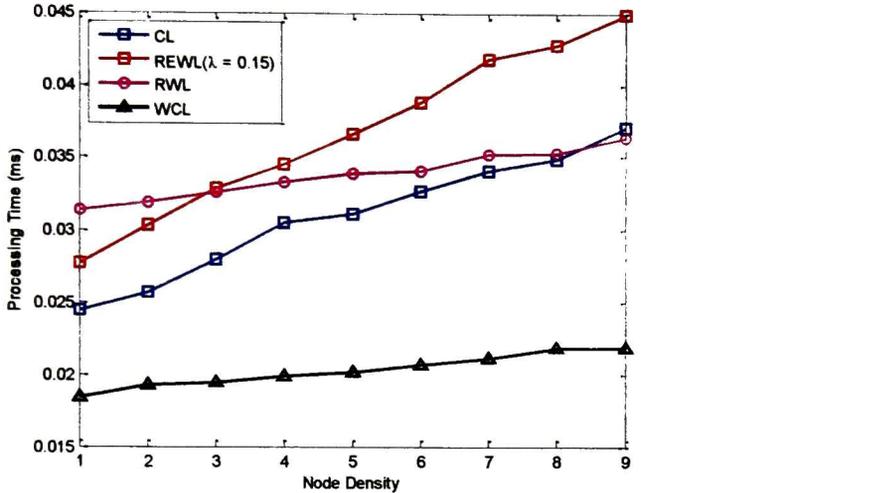


Fig. 3.20: Processing Time vs. Node Density for range-free algorithms.

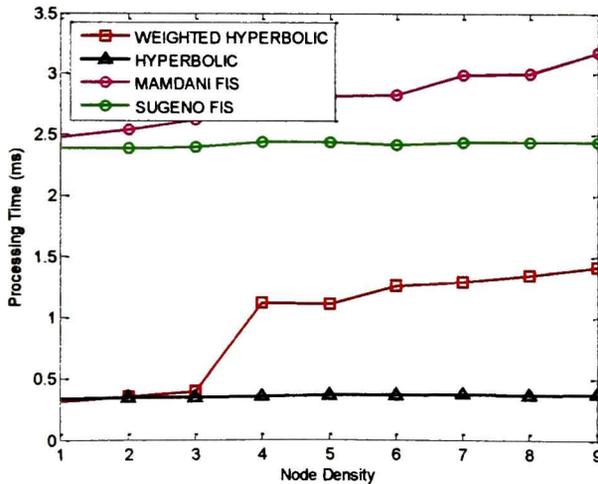


Fig. 3.21: Processing Time vs. Node Density for Mamdani & Sugeno FIS, Hyperbolic and Weighted Hyperbolic algorithms.

Analizando los algoritmos CL, WCL, RWL y REWL, se puede decir que estos algoritmos son complejidad de orden n . De las operaciones elementales que realiza cada algoritmo, se analiza la expresión que calcula el peso para cada algoritmo, ya que la expresión para calcular el centroide es similar para todos los algoritmos, donde el peso es la variante de

cada algoritmo. El algoritmo CL es el menos complejo, ya que sólo realiza una asignación como operación, después le sigue el algoritmo WCL, el cual realiza una división y una potencia, la cual depende del parámetro g . Los algoritmos RWL y REWL son más complejos que los algoritmos CL y WCL, ya que éstos realizan el algoritmo de búsqueda del valor máximo y mínimo de un arreglo de datos, el cual es de orden n , pero el algoritmo REWL es más complejo que el algoritmo RWL, ya que éste realiza dos restas y una potencia, donde la potencia es una operación con exponente flotante, en cambio el algoritmo RWL realiza una resta y una división.

En la Figura 3.22 se observa el desempeño de los algoritmos basados en distancia en cuanto a tiempo de procesamiento promedio requerido durante la estimación de un NOI. Analizando cada algoritmo de localización, se observa que el algoritmo hiperbólico es el de menor complejidad, ya que éste estima la posición de un nodo en una sola iteración, en cambio el algoritmo hiperbólico ponderado es de mayor complejidad ya que involucra más operaciones matemáticas, como multiplicación de matrices e inversiones. Se observa que los algoritmos de posicionamiento circular y multilateración WLS son los de mayor complejidad, ya que estos algoritmos son recursivos. Comparando estos dos algoritmos, multilateración WLS es más complejo que el algoritmo de posicionamiento circular, ya que éste involucra operaciones matriciales en cada iteración, en cambio, el algoritmo circular utiliza el método del gradiente descendiente para resolver el problema de localización, el cual involucra operaciones elementales sin el uso de operaciones matriciales como multiplicaciones e inversiones.

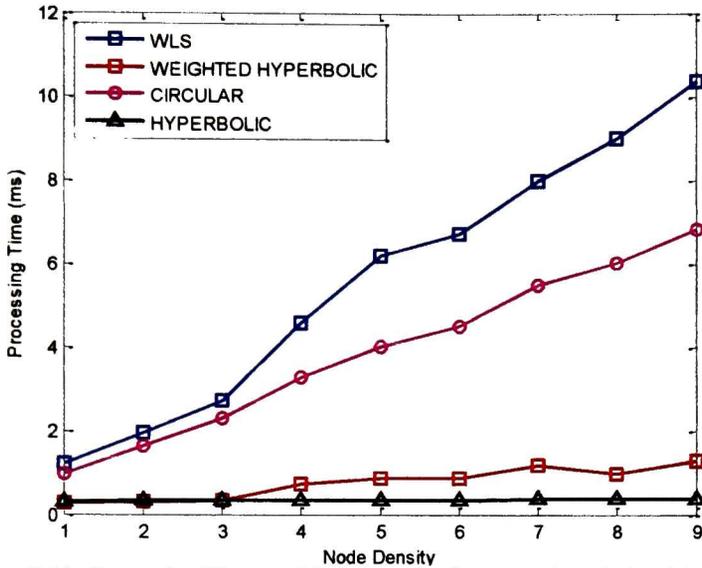


Fig. 3.22: Processing Time vs. Node Density for range-based algorithms.

3.7.2 Complejidad Computacional basada en el número de Operaciones Elementales

En esta sección se analiza la complejidad computacional en base al número de operaciones por segundo de los algoritmos de localización libres de distancia y basados en distancia para diferentes densidades de nodos. El número de operaciones de un algoritmo se refiere al número de sumas, restas, multiplicaciones y divisiones que consume un algoritmo durante su ejecución. Para obtener el número de operaciones elementales de un algoritmo, se realiza un conteo del número de ocurrencias de cada operación básica y se almacena en su respectiva variable.

A continuación se muestra un ejemplo sencillo por medio de ciclos anidados, que permite obtener el número de operaciones elementales de un algoritmo durante su ejecución.

Como se observa en el siguiente ejemplo, las variables de contadorSumas, contadorRestas, contadorMultiplicaciones y contadorDivisiones se utilizan como contadores del número de operaciones básicas.

```

for(int i = 0; i < N; i++){
    for(int j = 0; i < N; j++){
        x[i] = x[i] + (y[j]-z[j])/(y[i]*z[j]);
        contadorSumas = contadorSumas + 1;
        contadorRestas = contadorRestas + 1;
        contadorMultiplicaciones = contadorMultiplicaciones + 1;
        contadorDivisiones = contadorDivisiones + 1;
    }
}
    
```

En la Tabla II se observa el número de operaciones elementales de los algoritmos de localización libres de distancia para densidad de nodos de 1 y 2 nodos/10000m².

Tabla II. Número de operaciones elementales de los algoritmos libres de distancia para densidad de nodos de 1 y 2 nodos/(10,000m²).

Operación	Densidad de nodos							
	1				2			
	CL	WCL	RWL	REWL	CL	WCL	RWL	REWL
Sumas	15	15	15	55	30	30	30	120
Restas	--	--	6	50	--	--	11	110
Multiplicaciones	10	15	10	174	20	30	20	389
Divisiones	2	7	7	86	2	12	12	191
Total	27	37	38	365	52	72	73	810

Como se observa en la Tabla II, el algoritmo REWL es el algoritmo con mayor complejidad, ya que a medida que aumenta la densidad de nodos, el número de operaciones realizadas por este algoritmo durante su ejecución es mayor que los demás algoritmos mostrados en la Tabla II.

En la Tabla III se observa el número de operaciones elementales de los algoritmos de localización libres de distancia para densidad de nodos de 5 y 9 nodos/10000m².

Al igual que en la Tabla II, se observa que el algoritmo REWL es el de mayor complejidad, ya que el número de operaciones realizadas por este algoritmo, es mucho mayor que los demás algoritmos, como se muestra en dicha Tabla.

Tabla III. Número de operaciones elementales de los algoritmos libres de distancia para densidad de nodos de 5 y 9 nodos/(10,000m²).

Operación	Densidad de nodos							
	5				9			
	CL	WCL	RWL	REWL	CL	WCL	RWL	REWL
Sumas	75	75	75	315	138	138	138	588
Restas	--	--	26	290	--	--	47	542
Multiplicaciones	50	75	50	1034	92	138	92	1937
Divisiones	2	27	27	506	2	48	48	947
Total	127	177	178	2145	232	324	325	4014

En la Tabla IV y V se muestra el número de operaciones básicas realizadas por los algoritmos basados en distancia para diferentes densidades de nodos. De acuerdo a los resultados obtenidos, se observa que el algoritmo WLS realiza mayor número de operaciones que los demás algoritmos, por lo tanto es el algoritmo de mayor complejidad computacional de los algoritmos basados en distancia mostrados en la Tabla IV y V.

Tabla IV. Número de operaciones elementales de los algoritmos basados de distancia para densidad de nodos de 1 y 2 nodos/(10,000m²).

Operación	Densidad de nodos							
	1				2			
	Hyper-bolic	W hyper-bolic	Circular	WLS	Hyper-bolic	W hyper-bolic	Circular	WLS
Sumas	60	180	860	1720	125	1560	2930	5710
Restas	16	16	1396	1310	36	36	2796	2640
Multiplicaciones	90	262	1542	2455	180	1867	4427	7260
Divisiones	2	10	1170	1185	2	20	2360	2370
Total	168	468	4968	6670	343	3483	12513	17980

A continuación se muestran las Figuras 3.23 y 3.24, donde se observa el número de operaciones por segundo para diferentes números de nodos de referencia en una red.

En la Figura 3.23 se observa el número de operaciones por segundo de los algoritmos basados en distancia. En dicha figura se observa que el algoritmo WLS es el más complejo

computacionalmente, resultando así mismo, el algoritmo hiperbólico el de menor complejidad computacional.

Tabla V. Número de operaciones elementales de los algoritmos basados de distancia para densidad de nodos de 5 y 9 nodos/(10,000m²).

Operación	Densidad de nodos							
	5				9			
	Hyper-bolic	W hyper-bolic	Circu-lar	WLS	Hyper-bolic	W hyper-bolic	Circu-lar	WLS
Sumas	320	27900	31318	47059	593	182712	189060	241098
Restas	96	96	6952	6588	180	180	12896	12256
Multiplicaciones	450	29782	36138	51933	828	189067	200863	252964
Divisiones	2	50	5886	5883	2	92	10948	10942
Total	868	57828	80294	111463	1603	372051	413767	517260

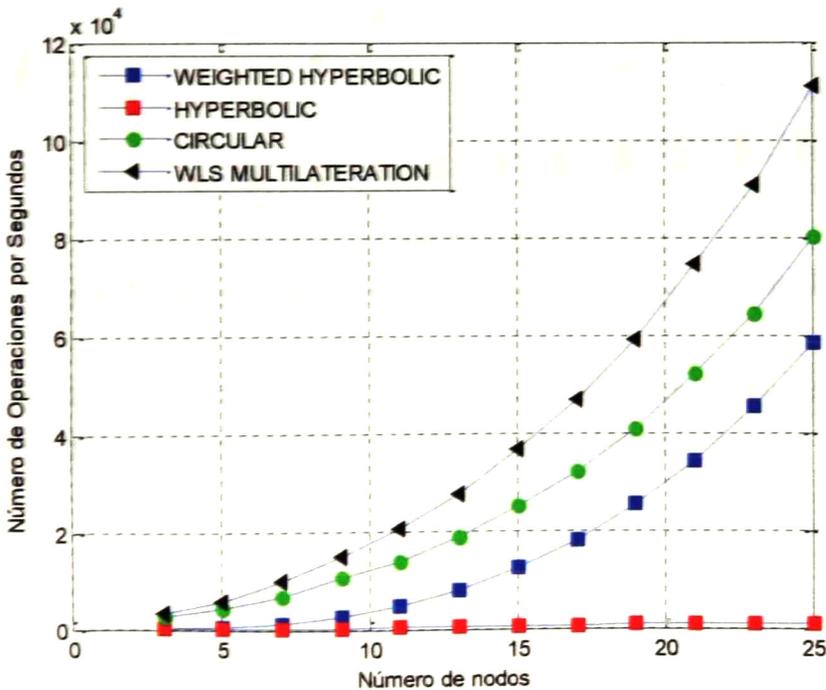


Fig. 3.23: Número de operaciones realizadas por los algoritmos basados en distancia.

En la Figura 3.24 se observa el número de operaciones por segundo de los algoritmos libres de distancia. En dicha figura se observa que el algoritmo REWL es el más complejo computacionalmente, resultando así mismo, el algoritmo CL el de menor complejidad computacional.

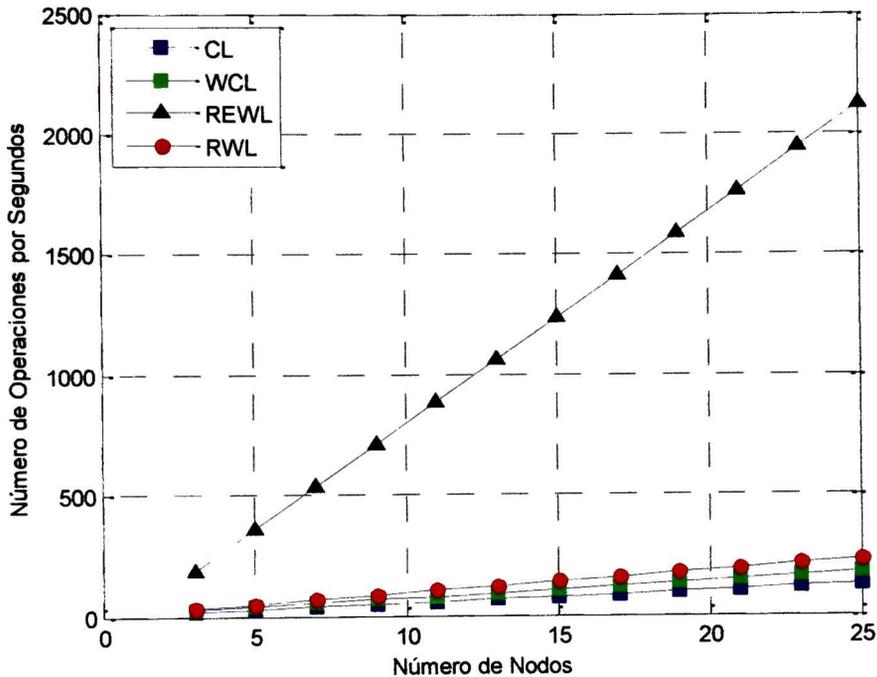


Fig. 3.24: Número de operaciones realizadas por los algoritmos libres de distancia.

Comparando las Figuras 3.23 y 3.24, se demuestra que los algoritmos basados en distancia son computacionalmente más costosos que los algoritmos libres de distancia.

Capítulo 4

Algoritmo de Multilateración por Mínimos Cuadrados Ponderados (WLS) Modificado

En este capítulo se propone una mejora del algoritmo de multilateración WLS, en cuanto a parámetros de desempeño de MSE y precisión, presentando mayor complejidad computacional, pero poco significativa, ya que el orden de complejidad es el mismo.

Este algoritmo está basado en el algoritmo de multilateración WLS, agregando una variante de ponderaciones, la cual está dada por una matriz de ponderaciones. Esta matriz de ponderaciones presenta un impacto sobre la matriz de covarianza de ruido gaussiano.

4.1 Descripción del Algoritmo

Partiendo de la expresión (2.13) del algoritmo de multilateración WLS, que representa el problema de optimización por mínimos cuadrados se observa la matriz de covarianza de ruido gaussiano, representada por K , la cual influye en el error de localización del NOI, ya que esta matriz contiene los valores de la varianza de ruido gaussiano, por lo que un ambiente muy ruidoso, este algoritmo puede ser muy ineficiente en cuanto a la precisión del NOI, incluso se pueden obtener errores muy grandes, lo cual implica que la localización del NOI esté fuera del área de cobertura de los nodos de referencia.

Asumiendo que las varianzas de ruido gaussiano son iguales, es decir:

$$K = \begin{bmatrix} \sigma_1^2 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_N^2 \end{bmatrix} \quad (4.1)$$

Donde K es la matriz de covarianza, cuya diagonal está compuesta por la varianzas de ruido gaussiano del NOI a los nodos de referencia i , donde $i = 1, 2, 3, \dots, N$.

Si las varianzas de ruido de gaussiano son iguales, implica que todos los nodos de referencia tienen la misma ponderación, es decir que dichos nodos de referencia tienen la misma probabilidad de estar más cerca o lejos del NOI, por lo que no se puede deducir a cual nodo de referencia se le debe asignar mayor ponderación.

Entonces, para resolver este problema se propone el uso de una matriz de ponderaciones denotada por \mathbf{R} , cuya matriz está compuesta por las distancias al cuadrado del NOI a los nodos de referencia \mathbf{i} , por lo tanto dicha matriz de ponderaciones, queda expresada por:

$$\mathbf{R} = \begin{bmatrix} r_1^2 & 0 & \dots & 0 \\ 0 & r_2^2 & \dots & 0 \\ \vdots & \vdots & \cdot & 0 \\ 0 & 0 & \dots & r_N^2 \end{bmatrix} \quad (4.2)$$

donde \mathbf{R} es la matriz de distancias al cuadrado del NOI a los nodos de referencia \mathbf{i} , cuya diagonal está compuesta por las distancias al cuadrado, es decir r_i^2 . Esta matriz \mathbf{R} es un término que sólo afecta a las varianzas de ruido gaussiano del NOI a los nodos de referencia \mathbf{i} . es decir, como los valores de la diagonal de la matriz de covarianza \mathbf{K} son iguales, entonces esta matriz \mathbf{R} se multiplica por la matriz de covarianza \mathbf{K} quedando una nueva matriz denotada por \mathbf{KR} cuya diagonal está compuesta por el término $(\sigma_i r_i)^2$

Con esta modificación hecha cada elemento de la diagonal de la matriz \mathbf{KR} depende de la distancia del NOI al nodo de referencia \mathbf{i} , por lo tanto los elementos de la diagonal principal presentan diferente valor, o bien ponderación, ya que la distancia r_i es diferente para cada nodo de referencia \mathbf{i} al NOI.

La expresión (2.13) sufre una modificación, quedando de la siguiente forma:

$$\hat{\mathbf{q}} = \underset{\mathbf{q}}{\operatorname{argmin}} [\mathbf{r} - \mathbf{f}(\mathbf{q})]^H (\mathbf{KR})^{-1} [\mathbf{r} - \mathbf{f}(\mathbf{q})] \quad (4.3)$$

Como se observa en la expresión anterior el término K se reemplaza por el término KR , analizando esta matriz, cada elemento de la diagonal principal posee diferente ponderación, ya que el NOI no es equidistante al nodo de referencia i . El concepto de menor o mayor ponderación se deduce del término $(KR)^{-1}$ de la expresión (4.3), por lo que si el término $(\sigma_i r_i)^2$ es mayor que los otros elementos de la diagonal principal implica que la distancia r_i es mayor que las otras distancias restantes, ya que σ_i presenta el mismo impacto sobre los nodos de referencia i , por lo tanto a mayor distancia menor ponderación o viceversa, es decir si $(\sigma_i r_i)^2$ es muy grande, al hacer la inversión de la matriz KR este valor será muy pequeño, por lo que la ponderación es muy baja, en caso contrario si $(\sigma_i r_i)^2$ es muy pequeño la ponderación es muy alta. Al indicar que la ponderación es baja o alta no implica que el valor es muy pequeño o muy grande, sino que el nodo de referencia i está más cerca o más lejos al NOI respecto a los otros nodos de referencia.

Desarrollando la expresión (4.3), se llega a la siguiente expresión como solución parcial del problema:

$$\hat{q} = q_0 + (D^H R^{-1} K^{-1} D)^{-1} D^H R^{-1} K^{-1} [r - f(q_0)] \quad (4.4)$$

Reordenado términos en la expresión (4.4), y como el algoritmo de multilateración WLS es iterativo se llega a la siguiente expresión como solución única del problema:

$$\hat{q}(n+1) = \hat{q}(n) + (D^H (KR)^{-1} D)^{-1} D^H (KR)^{-1} [r - f(\hat{q}(n))] \quad (4.5)$$

donde: $R = \text{diag}\{r_1^2, \dots, r_N^2\}$ como se observa en la expresión anterior, el algoritmo original sufre una modificación en el término K reemplazado por el término KR .

El objetivo principal de este algoritmo es ver el impacto que tiene la matriz KR , sobre el algoritmo, por lo que con esta modificación realizada es posible conocer cual nodo de referencia i se ve más afectado por el ruido gaussiano, lo cual se determina con el factor de ponderación discutido anteriormente.

4.2 Análisis de Desempeño del Algoritmo

Este algoritmo de localización fue evaluado en el mismo escenario que el algoritmo de multilateración WLS, considerando la misma área, las 10,000 iteraciones y el mismo modelo de propagación. A continuación se muestran unas gráficas de desempeño, del comportamiento de este algoritmo respecto al algoritmo de multilateración WLS.

En la Figura 4.1 se observa que el algoritmo de multilateración WLS presenta mejor desempeño de MSE y SD para niveles bajos de ruido, sin embargo, el algoritmo modificado presenta mejor desempeño para niveles altos de desviación estándar de ruido gaussiano. El desempeño de este algoritmo no es apreciable en este esquema, ya que la densidad de nodos es baja, lo cual repercute en el algoritmo de multilateración WLS modificado.

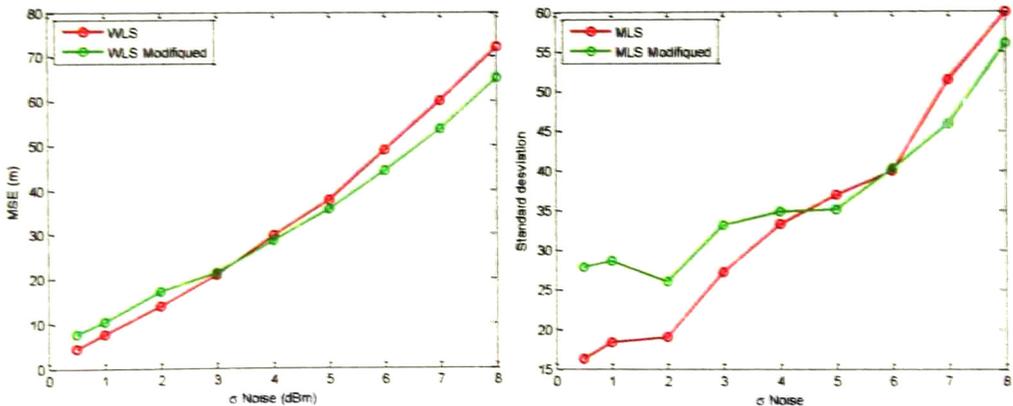


Fig. 4.1: MSE and SD vs. (σ Noise) for Node Density = 1.

En las Figuras 4.2 y 4.3 donde la densidad de nodos es más alta, se aprecia un mejor desempeño del algoritmo de multilateración WLS modificado. Como se observa en la Figura 4.2 existe una ligera diferencia de MSE y SD entre los dos algoritmos analizados, aunque el algoritmo de multilateración WLS presenta una mejor SD para niveles bajos de ruido gaussiano.

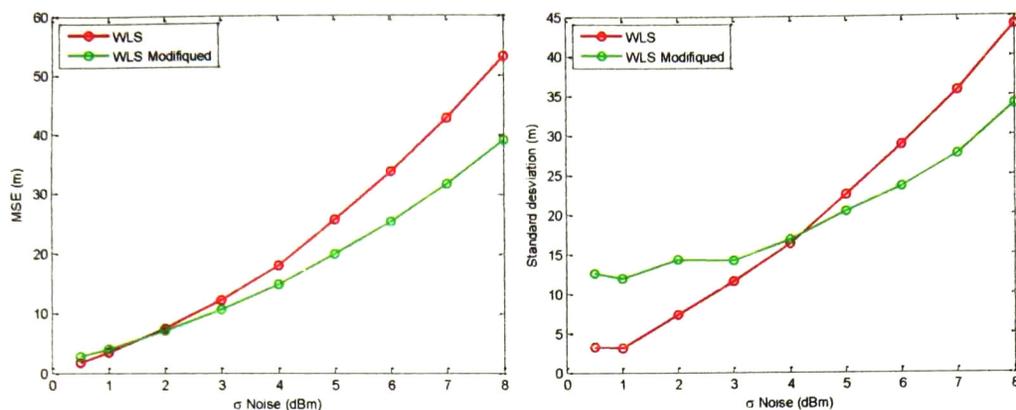


Fig. 4.2: MSE and SD vs. (σ Noise) for Node Density = 2.

Finalmente, en la Figura 4.3 se observa que el algoritmo de multilateración WLS modificado supera con creces al algoritmo de multilateración WLS en desempeño, sin embargo esta gran diferencia no se aprecia en la Figura 4.1 y 4.2; esto se debe a que la densidad de nodos es baja, por lo que el área de cobertura de los nodos de referencia no está bien definida y además no se tiene una gran cantidad de información sobre las distancias conocidas entre los nodos de referencia y el NOI, lo cual implica que también no se tenga la suficiente información para saber que tan lejos o cerca está el NOI respecto a los nodos de referencia. Por lo tanto, a mayor densidad de nodos se aprecia un mejor desempeño de este algoritmo modificado sobre el algoritmo original.

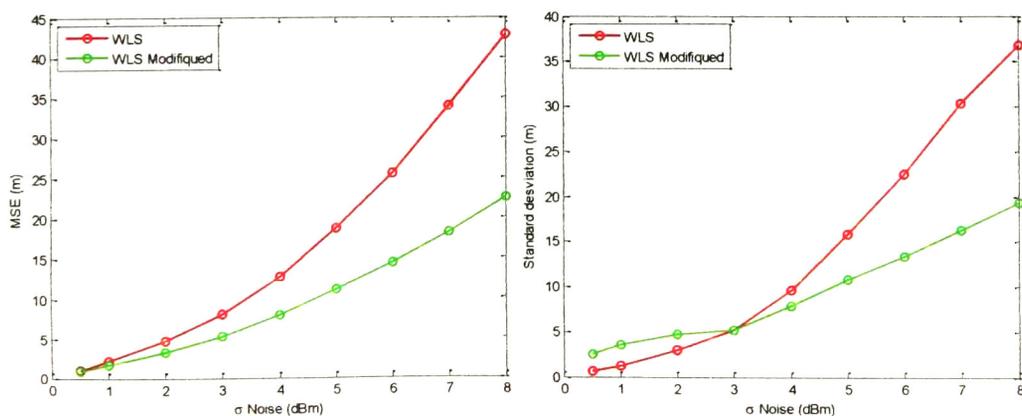


Fig. 4.3: MSE and SD vs. (σ Noise) for Node Density = 5.

Además se puede apreciar que para densidades altas de nodos el comportamiento de este algoritmo es más robusto que el algoritmo original, ya que conforme aumenta el nivel de ruido gaussiano, el MSE y la SD crecen con mayor celeridad en el algoritmo de multilateración WLS.

A continuación se muestran unos esquemas donde se varía la densidad de nodos en la red, para cierto nivel de ruido. Como se observa en la Figura 4.4 el algoritmo modificado presenta mejor desempeño de MSE y SD, ya que en las gráficas anteriores se muestra que para niveles mayores de desviación estándar de ruido gaussiano, el algoritmo modificado presenta mejor desempeño, alcanzando valores de MSE y SD mejores que el algoritmo de multilateración WLS, conforme aumenta la densidad de nodos.

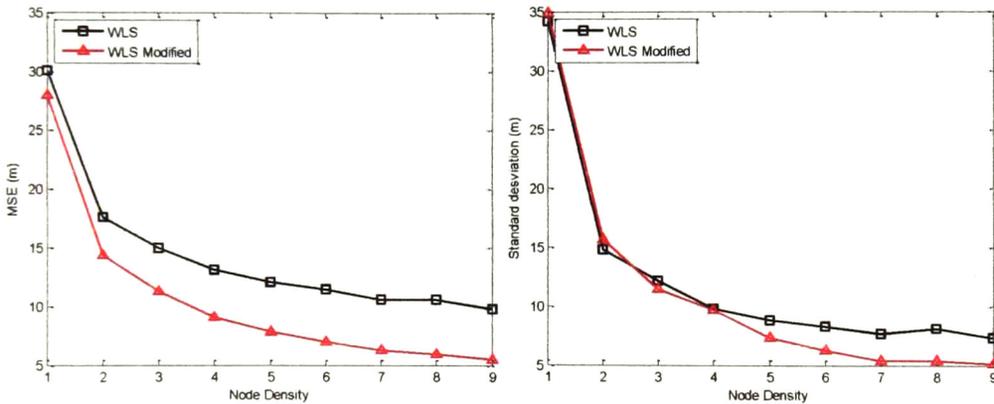


Fig. 4.4: MSE and SD vs. Node density for (σ Noise = 4 dBm).

En la Figura 4.5 el algoritmo de multilateración WLS se desempeña mejor en cuanto a MSE y SD, ya que el nivel de ruido gaussiano es bajo. Independientemente de la densidad de nodos, multilateración WLS presenta mejor desempeño que el algoritmo modificado para niveles bajos de ruido gaussiano, en caso contrario donde el ruido es alto, el algoritmo modificado se desempeña mucho mejor.

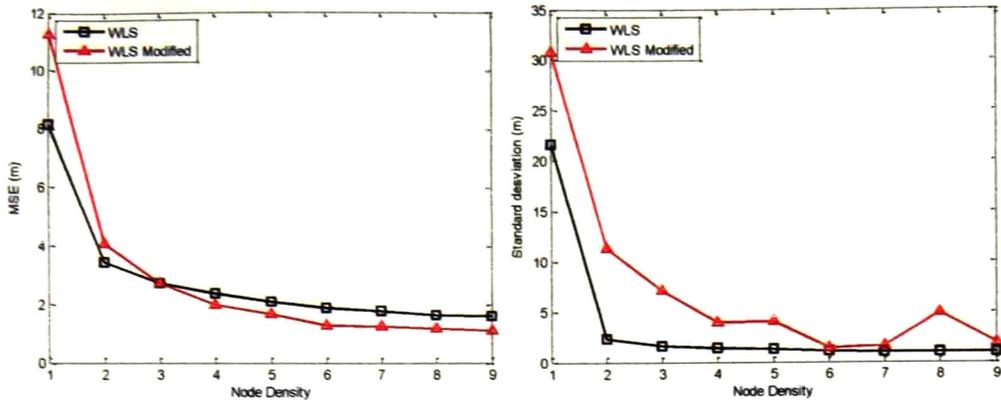


Fig. 4.5: MSE and SD vs. Node density for (σ Noise = 1 dBm).

Por último se analizan los esquemas de la CDF para ambos algoritmos analizados, donde se aprecia con mejor claridad la precisión de estos algoritmos.

En la Figura 4.6, se observa que el algoritmo de multilateración WLS presenta mejor precisión, ya que este algoritmo se desempeña mejor para niveles de ruido bajo, en el caso del esquema 4.7, el algoritmo modificado presenta una ligera mejora, ya que esta curva fue evaluada para una desviación estándar de ruido gaussiano mayor (4dBm).

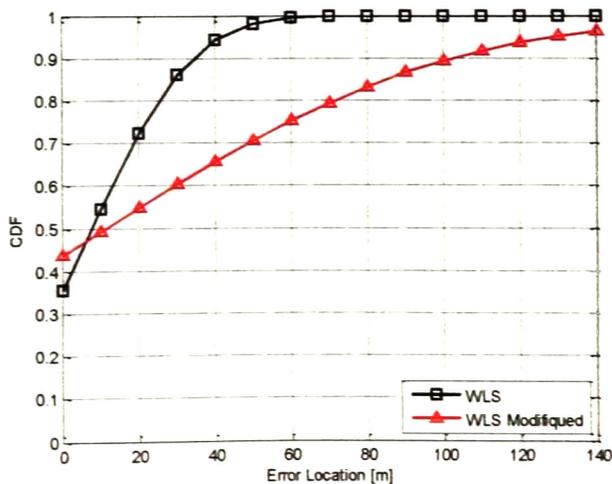


Fig. 4.6: CDF vs. Error Location for (σ Noise = 1 dBm and Node Density = 1).

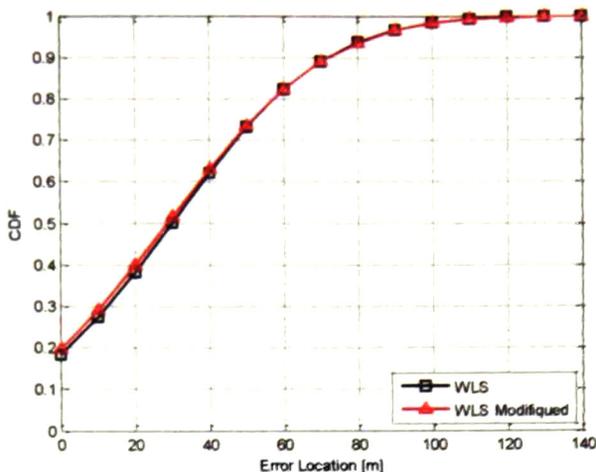
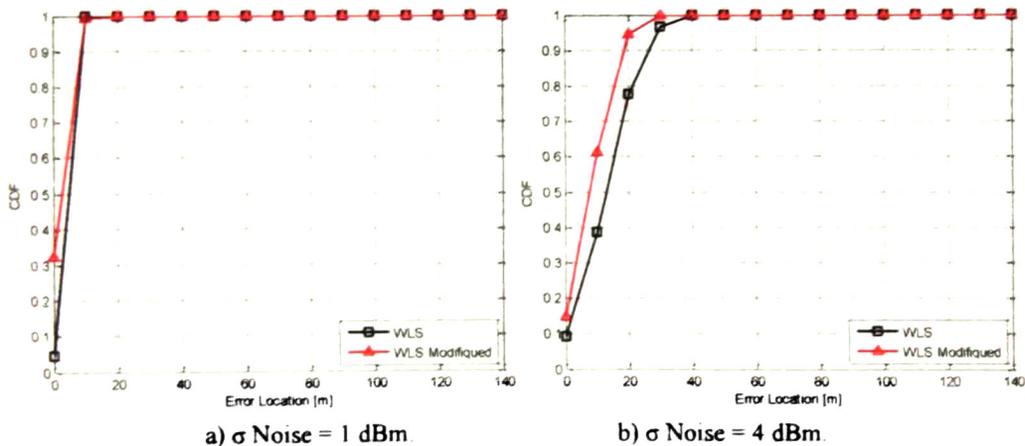


Fig. 4.7: CDF vs Error Location for (σ Noise = 4 dBm and Node Density = 1).

Ahora se analizan las Figuras 4.8 y 4.9, donde la densidad de nodos es mayor, dado que las desviaciones estándar de ruido gaussiano son de 1 dBm, 4 dBm, y 6 dBm en cada esquema de evaluación.



a) σ Noise = 1 dBm.

b) σ Noise = 4 dBm.

Fig. 4.8: CDF vs Error Location for Node Density = 5.

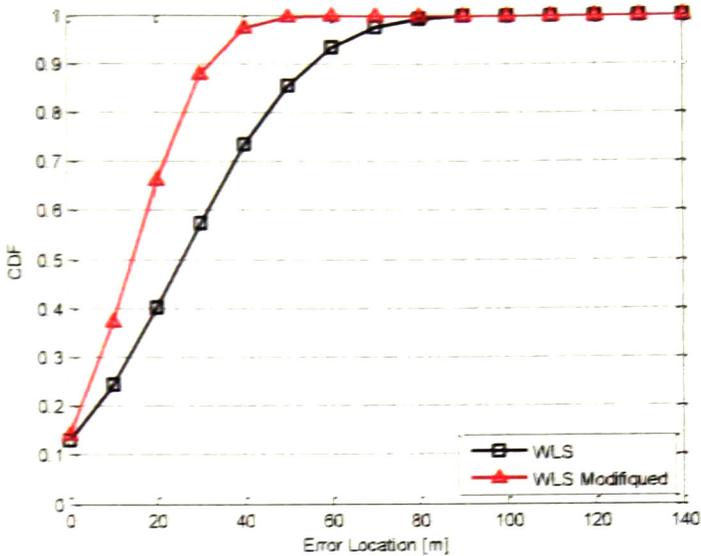


Fig. 4.9: CDF vs. Error Location for (σ Noise = 6 dBm and Node Density = 5).

En la Figura 4.8, para una densidad de nodos de 5, se presentan dos casos con niveles de ruido de 1dBm en el caso a) y 4dBm en el caso b). En el caso a) el algoritmo de multilateración WLS presenta una ligera mejora de precisión, en el caso b), donde la desviación estándar de ruido gaussiano es mayor, el algoritmo modificado, presenta mejor desempeño, para errores de localización menores de 40m, considerando este escenario de evaluación.

Finalmente en la Figura 4.9 se aprecia una gran diferencia de los algoritmos analizados, donde se observa que el algoritmo de multilateración WLS modificado, presenta en el 90% de los casos, errores de localización por debajo de los 30m, y en el 50% de los casos presenta errores menores de los 15m, en cambio el algoritmo original presenta errores menores de 60m en el 90% de los casos y de 25m en el 50% de los casos.

Por lo tanto el algoritmo WLS modificado es más preciso en ambientes muy ruidosos, comparado con el algoritmo de multilateración WLS, lo cual es benéfico, ya que este algoritmo modificado tiende a comportarse más robusto, lo cual garantiza que no haya errores de localización muy grandes o valores anormales.

También este algoritmo modificado es más robusto que el algoritmo WLS para altas densidades de nodos. El aumento de densidad de nodos es un factor que repercute de manera positiva en la precisión del algoritmo de multilateración WLS modificado sobre el algoritmo original, ya que los resultados obtenidos así lo demuestran.

Capítulo 5

Conclusiones y Trabajo Futuro

5.1 Conclusiones

Las principales conclusiones de este trabajo, de acuerdo a los resultados obtenidos son:

1. La localización en redes WSN, es altamente dependiente del escenario de aplicación, es decir, un escenario donde los nodos de referencia son distribuidos aleatoriamente implica mayor error en la localización, comparado con un escenario donde los nodos de referencia son distribuidos en forma de retícula. Esto se debe a que en el primer escenario existe la probabilidad de que el nodo de interés sea localizado fuera de la cobertura de los nodos ancla, ocasionado por niveles altos de ruido gaussiano, debido a que este escenario fue simulado en un área grande, en cambio en el segundo escenario, donde el área de cobertura es más pequeña y los nodos de referencia están más cercanos uno del otro, el error de localización es menor, ya que la zona de cobertura es más pequeña.
2. Los algoritmos libres de distancia son menos complejos que los algoritmos basados en distancia, pero tienden a ser más imprecisos en la localización, por lo que no es recomendable su aplicación en escenarios de áreas grandes, ya que los errores de localización son considerables respecto a los algoritmos basados en distancia.
3. Los algoritmos basados en distancia son altamente dependientes de la cobertura de los nodos ancla, por lo que este tipo de algoritmos funcionan para tres o más nodos de referencia, de lo contrario, el nodo de interés puede ser localizado fuera de la zona de cobertura, lo cual implica un error grande en la localización.
4. Se confirmó que la densidad de nodos es un factor importante en la localización, ya que mayor número de nodos en el escenario de localización implica que el nodo de interés

obtenga mayor información de sus nodos vecinos, lo que propicia una mejor precisión en la localización, pero, el algoritmo se vuelve más complejo.

5. La técnica de localización de KNN con fingerprints sólo es aplicable en escenarios donde los nodos de referencia son distribuidos de forma reticular, ya que de esta forma se pueden tomar muestras de niveles de potencia dentro de un área ya definida, en cambio en un escenario donde los nodos de referencia son móviles, no existe un área común.
6. El uso de los algoritmos de centroide con FIS no es recomendable, ya que dependiendo del escenario de aplicación el algoritmo tendría que modificar su estructura, es decir, se tiene que añadir mayor número de variables lingüísticas, además no existe una regla para determinar las funciones de pertenencia y el número de estas, por lo que sólo pueden ser ajustadas dependiendo del escenario.
7. Se calculó el número de operaciones elementales (sumas, restas, multiplicaciones y divisiones) de los algoritmos de localización, con el fin de determinar el algoritmo más complejo en cuanto a procesamiento, de donde se probó que los algoritmos de localización basados en distancia son computacionalmente más complejos que los algoritmos libres de distancias.

5.2 Contribuciones

Como contribuciones del trabajo realizado cabe destacar:

1. Se realizó un análisis comparativo de las métricas de desempeño de los algoritmos de localización basados en distancia y libres en distancia, donde se demostró que los algoritmos basados en distancia presentan mejor precisión que los algoritmos libres de distancia para niveles bajos de ruido gaussiano de acuerdo a los resultados obtenidos, pero, estos algoritmos son más complejos computacionalmente que los algoritmos libres de distancia.

2. Se desarrolló una modificación al algoritmo de Multilateración WLS lo cual mejoró el desempeño del algoritmo original, mostrando un mejor MSE y precisión para el escenario de evaluación propuesto, así mismo mejor robustez para valores altos de desviación estándar de ruido gaussiano.

5.3 Trabajo Futuro

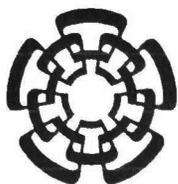
1. Evaluar las métricas de desempeño de los algoritmos de localización en escenarios multi-salto, es decir establecer la comunicación entre varios nodos en una red WSN, a través de múltiples saltos hasta localizar al nodo de interés.
2. Evaluar los algoritmos basados en distancia en escenarios multi-salto, con el fin de determinar las métricas de desempeño.
3. Evaluar los algoritmos de localización centralizados y distribuidos en un ambiente real, donde se tenga un área con varios nodos de referencia con una posición definida y varias muestras de los niveles de potencia empleando el método de fingerprint, con el fin de determinar el patrón de radiación del ambiente de evaluación para poder aplicar el algoritmo de localización.
4. Estudiar el desempeño de otros métodos de localización como ToA, TDoA o AoA, con el fin de determinar su impacto en una red WSN, respecto a los algoritmos basados en RSS.

Referencias Bibliográficas

- [1] Omotayo G. Adewumi, Karim Djouani, Anish M. Kurien. (February 2013). “*RSSI Based Indoor and Outdoor Distance Estimation for Localization in WSN.*” IEEE International Conference on, Cape Town, pp. 1534-1539.
- [2] Adel, G.A.E., Hussein, A.E., Salwa, E.R., Magdy, M.I. (2010). “*An Energy Aware WSN Geographic Routing Protocol.*” Universal Journal of Computer Science and Engineering Technology, pp. 105-111.
- [3] Neal Patwari, Joshua N. Ash, Spyros Kyperountas, Alfred O. Hero III, Randolph L. Moses, and Neiyer S. Correal. (July 2005). “*Locating the nodes: cooperative localization in wireless sensor networks.*” IEEE Signal Processing Magazine, vol. 22, no. 4, pp. 54–69.
- [4] G. Mao, B. Fidan and B. D. O. Anderson, (July 2007). “*Wireless sensor network localization techniques.*” Computer Networks. NY, USA, vol. 51, issue 10, pp. 2529-2553.
- [5] L.A. Castro and J. Favela. (September 2005). “*Continuous tracking of user location using recurrent neural networks.*” in Proc. 6th Mexican Int. Conf. Computer Science. Puebla, Mexico, pp. 174–181.
- [6] V. Pérez-González**, R. Torres-Villegas**, D. Muñoz Rodríguez and C. Vargas-Rosales. (2012). “*Relational Position Location in Ad Hoc Networks.*”
- [7] David Munoz, Frantz Bouchereau, Cesar Vargas and Rogerio Enriquez. 2009, “*Position Location Techniques and Applications.*” Academic Press. pp. 87-89.
- [8] Thompson, B. and R. M. Buehrer. (2012). “*Characterizing and improving the collaborative position location problem.*” IEEE Workshop on Positioning Navigation and Communication (WPNC), pp. 42-46.
- [9] Ali H. Sayed, Alireza Tarighat and Nima Khajehnouri. (July 2005). Network-Based Wireless Location: Challenges faced in developing techniques for accurate wireless location information, IEEE Signal Processing Magazine, pp 24-40.
- [10] Rappaport, T.S.: Wireless Communications: Principles and Practice. Second edn. Prentice-Hall, New Jersey (2002).

- [11] Kuang Xing-Hong, Shao Hui-He, Feng Rui. (March 2008). "*A New Distributed Localization Scheme for Wireless Sensor Network*." Acta Automatica Sinica. Vol. 34, issue 3, pp. 344-348.
- [12] Xiang Ji, Hongyuan Zha. (March 2004). "*Sensor Positioning in Wireless Ad-hoc Sensor Networks Using Multidimensional Scaling*." IEEE INFOCOM 2004. pp. 2652-2661.
- [13] Dragos Niculescu, Badri Nath. (November 2001). "*Ad Hoc Positioning System (APS)*." IEEE GLOBECOM 2001. vol.5, San Antonio, TX, pp. 2926-2931.
- [14] Liu, b., Lin, K. (January 2008) "*Distance Difference Error Correction by Least Square for Stationary Signal-Strength-Difference-Based Hyperbolic Location in Cellular Communications*." Proceedings On Vehicular Technology, IEEE Transactions, vol. 57, issue 1, pp. 227-238.
- [15] T. He, C. Huang, B.M. Blum, J. A. Swnkovic, and T. Abdelzaher. (September 2003). "*Range-Free Localization Schemes for Large Scale Sensor Networks*." Proceedings of the ninth annual international conference on Mobile computing and networking (MOBICOM 2003). San Diego, California, pp. 81-95.
- [16] Eva M. García, Aurelio Bermúdez, Rafael Casado, Francisco J. Quiles. (September 2007). "*Wireless Sensor Network Localization using Hexagonal Intersection*." Proceedings of the IFIP WG 6.8 First International Conference on Wireless Sensor and Actor Networks (WSAN 2007). Albacete, Spain, vol. 248; pp. 155-166.
- [17] N. Bulusu, J. Heidemann and D. Estrin. (October 2000). "*GPS-less Low-Cost Outdoor Localization for Very Small Devices*." IEEE Personal Communications Magazine. Los Angeles, USA, Vol. 7, No. 5, pp. 28-34.
- [18] Blumenthal, J., Grossmann, R., Golatowski, F., Timmermann, D. (October 2007). "*Weighted Centroid Localization in Zigbee-based Sensor Networks*." Proceedings of the IEEE International Symposium on Intelligent Signal Processing (WISP). Alcala de Henares, pp. 1-6.
- [19] Schuhmann, S., Herrmann, K., Rothermel, K., Blumenthal, J., Timmermann, D. (2008). "*Improved Weighted Centroid Localization in Smart Ubiquitous Environments*." In: Ubiquitous Intelligence and Computing: Proceedings of the 5th International Conference (UIC). Vol. 5061, Springer Berlin / Heidelberg, pp. 20-34.

- [20] Christine Laurendeau and Michel Barbeau (November 2009). "*Centroid Localization of Uncooperative Nodes in Wireless Networks Using a Relative Span Weighting Method.*" In: EURASIP Journal on Wireless Communications and Networking. pp. 358-361.
- [21] Y. Sukhyun, L. Jaehun, and Wooyong. (May 2009). "*A soft computing approach to localization in wireless sensor networks.*" Expert Systems with Applications. Vol. 36, issue 4, pp. 7552-7561.
- [22] Yun, S., Lee, J., Chung, W., & Kim, E. (September 2005). "*Centroid localization method in wireless sensor networks using TSK fuzzy modeling.*" International symposium on advanced intelligent systems, Sokcho, Korea, pp. 971-974.
- [23] Mamdani, E.H. (December 1977). "*Applications of fuzzy logic to approximate reasoning using linguistic synthesis.*" IEEE Transactions on Computers. pp. 1182-1191.
- [24] P. Wolf. (October 2007). "*Fuzzy Logic in Decision Making Process.*" Proceedings of 3rd WSEAS/IASME International Conference on Dynamical Systems and Control (CONTROL 2007). Arcachon, France, pp. 282-287
- [25] M. Sugeno. (July-August 1985). "*An introductory survey of fuzzy control.*" Information Science. Vol. 36, issues 1-2, pp. 59-83.
- [26] I. Ianchu and C. I. Popirlan. (June 2010) "*Mamdani Fuzzy Logic Controller with Mobile Agents for Matching.*" Proceedings of Recent Advances in Neural Networks, Fuzzy Systems & Evolutionary Computing (NNECFS'10). Iasi, Romania, pp. 117- 112.
- [27] Azat Rozyyev, Halabi Hasbullah and Fazli Subhan. (2012). "*Combined K-Neighbors and Fuzzy Logic Indoor Localization Technique for Wireless Sensor Network.*" Research Journal of Information Technology. vol. 4, issue. 4, pp. 155-165.
- [28] R.A. Poisel. (2005). "*Electronic Warfare Target Location Methods.*" Norwood, MA: Artech House.
- [29] Tarrío, P., Bernardos, A., Besada, J. and Casar, J. (October 2008). "*A new positioning technique for RSS-Based localization based on a weighted least squares estimator.*" IEEE International Symposium on Wireless Communication Systems. Reykjavik, pp. 633-637.



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS DEL I.P.N. UNIDAD GUADALAJARA

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional aprobó la tesis

Evaluación del desempeño de localización geográfica en redes inalámbricas de sensores y actuadores

del (la) C.

Joaquin MASS SANCHEZ

el día 28 de Febrero de 2014.

Dr. Deni Librado Torres Román
Investigador CINVESTAV 3B
CINVESTAV Unidad Guadalajara

Dr. Ramón Parra Michel
Investigador CINVESTAV 3B
CINVESTAV Unidad Guadalajara

Dr. César Vargas Rosales
Director de la Carrera de
Microelectrónica y
Telecomunicaciones del Centro de
Electrónica y Telecomunicaciones
Instituto Tecnológico de Estudios
Superiores de Monterrey Campus
Monterrey



CINVESTAV - IPN
Biblioteca Central



SSIT0012324